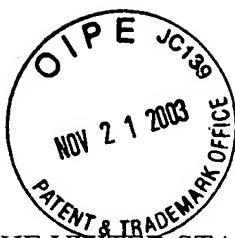


01807.002379.



PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:)	
	:	Examiner: Not Yet Assigned
FABRICE LE LEANNEC, ET AL.)	
	:	Group Art Unit: 2651
Application No.: 10/618,667)	
	:	
Filed: July 15, 2003)	
	:	
For: METHOD AND DEVICE FOR)	
PROCESSING A REQUEST OR	:	
COMPRESSED DIGITAL DATA)	November 20, 2003

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUBMISSION OF PRIORITY DOCUMENT

Sir:

In support of Applicants' claim for priority under 35 U.S.C. § 119, enclosed is
a certified copy of the following French application:

0208900, filed July 15, 2002.



Applicants' undersigned attorney may be reached in our New York office by telephone at (212) 218-2100. All correspondence should continue to be directed to our address given below.

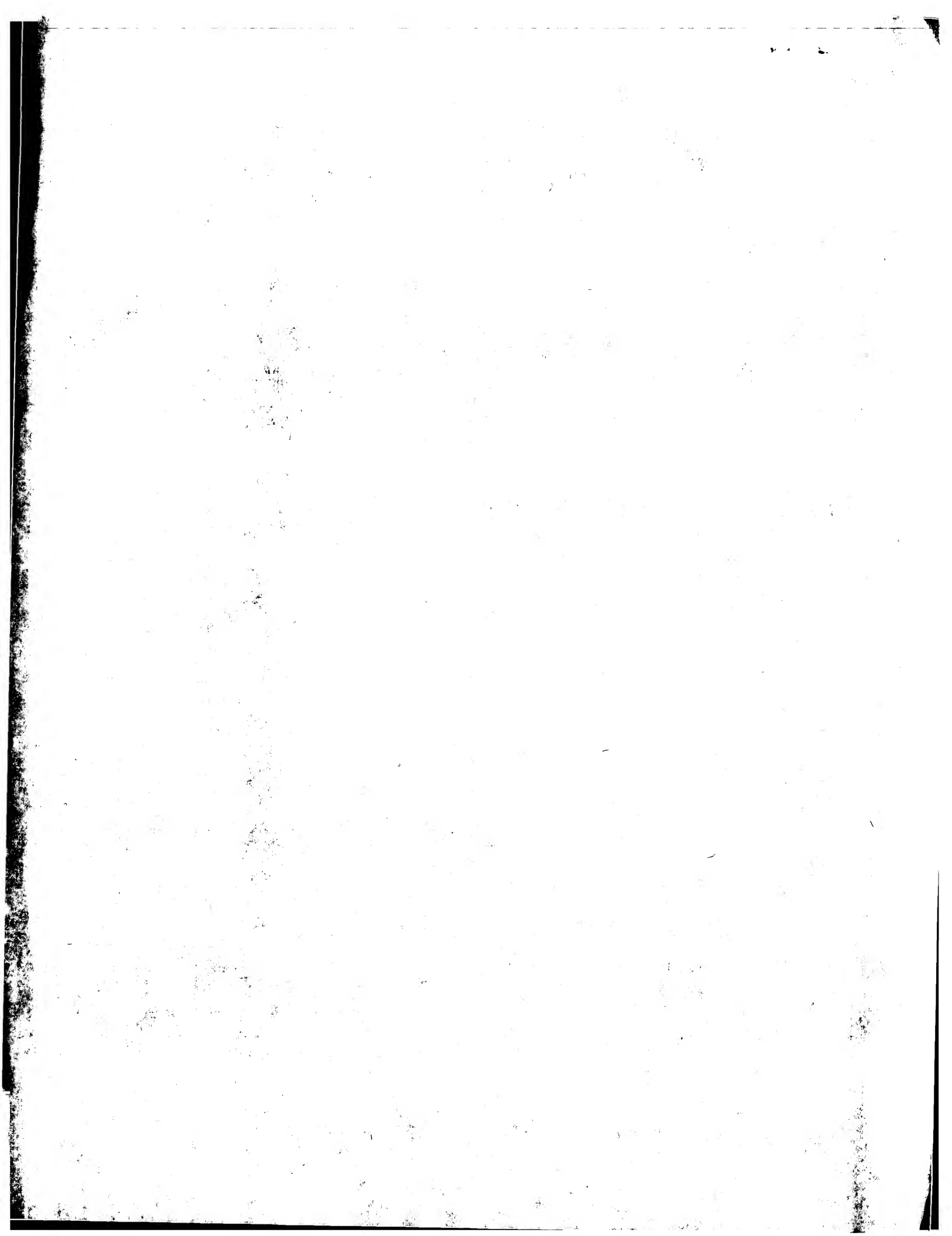
Respectfully submitted,

Raymond D. Perna
Attorney for Applicants

Registration No. 44,063

FITZPATRICK, CELLA, HARPER & SCINTO
30 Rockefeller Plaza
New York, New York 10112-3800
Facsimile: (212) 218-2200

NY_MAIN 390696v1





BREVET D'INVENTION

CERTIFICAT D'UTILITÉ - CERTIFICAT D'ADDITION

COPIE OFFICIELLE

Le Directeur général de l'Institut national de la propriété industrielle certifie que le document ci-annexé est la copie certifiée conforme d'une demande de titre de propriété industrielle déposée à l'Institut.

Fait à Paris, le 08 AOUT 2003

Pour le Directeur général de l'Institut
national de la propriété industrielle
Le Chef du Département des brevets

A handwritten signature in black ink, appearing to read 'M+Leuc', enclosed within a large, loopy oval stroke.

Martine PLANCHE

INSTITUT
NATIONAL DE
LA PROPRIÉTÉ
INDUSTRIELLE

SIEGE
26 bis, rue de Saint Petersburg
75800 PARIS cedex 08
Téléphone : 33 (0)1 53 04 53 04
Télécopie : 33 (0)1 53 04 45 23
www.inpi.fr

REQUÊTE EN DÉLIVRANCE
page 1/2



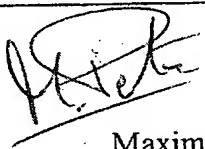
Cet imprimé est à remplir lisiblement à l'encre noire

DB 540 W / 300301

<p>REMISE DES PIÈCES</p> <p>DATE 15 JUL 2002</p> <p>LIEU 75 INPI PARIS</p> <p>N° D'ENREGISTREMENT 0208900</p> <p>NATIONAL ATTRIBUÉ PAR L'INPI</p> <p>DATE DE DÉPÔT ATTRIBUÉE 15 JUL. 2002</p> <p>PAR L'INPI</p>		<p>1 NOM ET ADRESSE DU DEMANDEUR OU DU MANDATAIRE À QUI LA CORRESPONDANCE DOIT ÊTRE ADRESSÉE</p> <p>RINUY, SANTARELLI 14, avenue de la Grande Armée 75017 PARIS</p>	
<p>Vos références pour ce dossier (facultatif) BIF023183/MP/MPA</p>			
<p>Confirmation d'un dépôt par télécopie</p> <p><input type="checkbox"/> N° attribué par l'INPI à la télécopie</p>			
<p>2 NATURE DE LA DEMANDE</p> <p>Demande de brevet <input checked="" type="checkbox"/></p> <p>Demande de certificat d'utilité <input type="checkbox"/></p> <p>Demande divisionnaire <input type="checkbox"/></p> <p>Demande de brevet initiale N° _____ Date _____</p> <p>ou demande de certificat d'utilité initiale N° _____ Date _____</p> <p>Transformation d'une demande de brevet européen Demande de brevet initiale N° _____ Date _____</p>		<p>Cochez l'une des 4 cases suivantes</p>	
<p>3 TITRE DE L'INVENTION (200 caractères ou espaces maximum)</p> <p>Procédé et dispositif de traitement d'une requête ou de données numériques compressées</p>			
<p>4 DÉCLARATION DE PRIORITÉ OU REQUÊTE DU BÉNÉFICE DE LA DATE DE DÉPÔT D'UNE DEMANDE ANTÉRIEURE FRANÇAISE</p>		<p>Pays ou organisation _____ N° _____</p> <p>Date _____</p> <p>Pays ou organisation _____ N° _____</p> <p>Date _____</p> <p>Pays ou organisation _____ N° _____</p> <p>Date _____</p> <p><input type="checkbox"/> S'il y a d'autres priorités, cochez la case et utilisez l'imprimé «Suite»</p>	
<p>5 DEMANDEUR</p> <p>Nom ou dénomination sociale</p> <p>Prénoms</p> <p>Forme juridique</p> <p>N° SIREN</p> <p>Code APE-NAF</p> <p>Adresse</p> <p>Rue</p> <p>Code postal et ville</p> <p>Pays</p> <p>Nationalité</p> <p>N° de téléphone (facultatif)</p> <p>N° de télécopie (facultatif)</p> <p>Adresse électronique (facultatif)</p>		<p><input type="checkbox"/> S'il y a d'autres demandeurs, cochez la case et utilisez l'imprimé «Suite»</p> <p>CANON KABUSHIKI KAISHA</p> <p>Société de droit Japonais</p> <p>30-2, Shimomaruko 3-chome, Ohta-ku,</p> <p>Tokyo</p> <p>JAPON</p> <p>JAPONAISE</p>	

BREVET D'INVENTION
CERTIFICAT D'UTILITÉ
REQUÊTE EN DÉLIVRANCE
 page 2/2

R2

REMISE DES PIÈCES DATE 15 JUIL 2002 LIEU 75 INPI PARIS N° D'ENREGISTREMENT 0208900 NATIONAL ATTRIBUÉ PAR L'INPI		Réservé à l'INPI		DB 540 W / 300301
Vos références pour ce dossier : <i>(facultatif)</i>		BIF023183/MP/MPA		
6 MANDATAIRE Nom Prénom Cabinet ou Société N° de pouvoir permanent et/ou de lien contractuel Adresse Rue Code postal et ville N° de téléphone <i>(facultatif)</i> N° de télécopie <i>(facultatif)</i> Adresse électronique <i>(facultatif)</i>		RINUY, SANTARELLI 14 AVENUE DE LA GRANDE ARMÉE 75017 PARIS 01 40 55 43 43		
7 INVENTEUR (S) Les inventeurs sont les demandeurs		<input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non Dans ce cas fournir une désignation d'inventeur(s) séparée		
8 RAPPORT DE RECHERCHE Établissement immédiat ou établissement différé		Uniquement pour une demande de brevet (y compris division et transformation) <input checked="" type="checkbox"/> Établissement immédiat <input type="checkbox"/> Établissement différé		
Paiement échelonné de la redevance		Paiement en deux versements, uniquement pour les personnes physiques <input type="checkbox"/> Oui <input type="checkbox"/> Non		
9 RÉDUCTION DU TAUX DES REDEVANCES		Uniquement pour les personnes physiques <input type="checkbox"/> Requête pour la première fois pour cette invention (joindre un avis de non-imposition) <input type="checkbox"/> Requête antérieurement à ce dépôt (joindre une copie de la décision d'admission pour cette invention ou indiquer sa référence):		
Si vous avez utilisé l'imprimé «Suite», indiquez le nombre de pages jointes				
10 SIGNATURE DU DEMANDEUR OU DU MANDATAIRE (Nom et qualité du signataire)		VISA DE LA PRÉFECTURE OU DE L'INPI		
Maxime PETIT N°00.0407 RINUY, SANTARELLI				

5

10 La présente invention concerne un procédé et un dispositif de traitement d'une requête provenant d'un premier appareil de communication connecté à travers un réseau de communication à un deuxième appareil de communication distant.

L'invention concerne également un procédé et un dispositif de
15 traitement de données numériques compressées reçues par le premier appareil de communication.

L'invention s'applique notamment dans le domaine du traitement des images compressées et, par exemple, dans le domaine du traitement d'images compressées conformes à la norme JPEG2000.

20 Selon cette norme, un signal numérique d'image compressé possède une structure générale comportant des données d'en-tête constituant un en-tête principal et un corps qui comporte, sous forme compressée, des données représentatives de grandeurs physiques que sont les pixels et qui sont regroupées en blocs de données (connus en terminologie anglo-saxonne sous
25 le terme "code-blocks") ordonnés dans le signal.

Le corps du signal correspond au moins à une tuile qui représente de façon compressée une partie rectangulaire du signal d'image d'origine. Chaque tuile est formée de données d'en-tête de tuile et d'un corps de tuile contenant les blocs de données compressés correspondant à la tuile considérée.

30 Plus particulièrement, le corps de chaque tuile comporte des paquets de données qui sont chacun constitués de données d'en-tête de paquet et d'un corps de paquet.

Le corps de paquet contient à son tour plusieurs blocs de données compressés et les données d'en-tête du paquet contiennent notamment une liste de tous les blocs contenus dans le corps de paquet.

5 Chaque bloc de données compressé est une représentation compressée d'une partie rectangulaire élémentaire du signal d'image qui a été transformé, de manière connue, en sous-bandes de fréquence.

Il convient de noter que chaque bloc de données est compressé sur plusieurs niveaux ou couches de qualité et chaque niveau ou couche de qualité d'un bloc se trouve dans un paquet distinct.

10 Chaque paquet de données d'un signal d'image conforme à la norme JPEG2000 contient donc un ensemble de blocs de données compressés correspondant chacun à une tuile, une composante (ex : luminance ou chrominance), un niveau de résolution, un niveau de qualité et une position spatiale (connue en terminologie anglo-saxonne sous le terme "precinct")
15 donnés.

Dans une architecture de communication de type client-serveur, l'utilisateur reçoit et stocke un signal numérique d'image compressé qui est, par exemple, conforme à la norme JPEG2000.

20 Le signal ainsi reçu est alors décompressé et restitué dans l'ordinateur client afin que l'utilisateur puisse utiliser le signal d'image ou une partie de celui-ci s'il le souhaite.

La transmission du fichier informatique contenant les données compressées peut prendre un temps relativement long si le fichier est volumineux.

25 En outre, la transmission d'un tel fichier peut être soumise à des perturbations risquant de rendre certaines données erronées.

Ce phénomène est accentué lorsque le fichier est volumineux.

De plus, il n'est pas rare que des problèmes liés au matériel informatique surviennent lors de la réception et du stockage d'un fichier de
30 données et que certaines des données soient perdues pour l'utilisateur, voire tout le fichier.

Il arrive également de temps en temps que des fichiers trop volumineux ne puissent tout simplement pas être exploités par l'utilisateur compte tenu des capacités de stockage et de traitement insuffisantes de l'ordinateur de l'utilisateur.

5 Par ailleurs, dans certaines situations, la totalité d'un signal numérique compressé qui, par exemple, est un signal d'image n'intéresse pas l'utilisateur.

L'utilisateur peut souhaiter s'intéresser, en effet, plus particulièrement à une partie du signal d'image ou bien à un niveau de résolution déterminé et non à tous les niveaux de résolution, ceci sans avoir besoin de décompresser
10 tout le signal.

Ainsi, pour différentes raisons, on souhaite recevoir seulement une partie du signal numérique compressé disponible au serveur.

Dans ce cas, le serveur doit rechercher dans le signal compressé les paquets de données correspondant à la partie du signal demandée par
15 l'utilisateur et les extraire de ce dernier avant de les transmettre à la machine client.

Cependant, cette recherche nécessite de décoder tout ou partie des données d'en-tête des paquets du corps du signal afin d'obtenir les informations sur les longueurs respectives élémentaires des contributions des blocs de
20 données contenus dans le corps de chaque paquet.

La longueur du corps de chaque paquet est déduite de ces longueurs élémentaires et il est ainsi possible de trouver la position du paquet suivant dans le corps du signal. Cette solution est d'autant plus longue et fastidieuse à mettre en œuvre qu'elle nécessite de parcourir tout ou partie du corps du signal
25 pour trouver les paquets recherchés afin de les extraire.

On notera que ce qui précède s'applique aussi bien avec un signal numérique d'image compressé conforme à la norme JPEG2000 qu'avec tout type de signal numérique compressé (signaux sonores, signaux vidéo, signaux de données issues d'un télécopieur ou d'autres appareils électroniques ...).

30 Il serait par conséquent utile de disposer d'un nouveau procédé et d'un nouveau dispositif permettant de traiter de manière plus rapide dans un appareil de communication une requête émanant d'un appareil de

communication distant et visant à obtenir une partie d'un signal numérique compressé.

La présente invention a ainsi pour objet un procédé de traitement d'une requête provenant d'un premier appareil de communication connecté à travers un réseau de communication à un deuxième appareil de communication distant, le procédé comportant une étape de réception de la requête d'obtention de données numériques d'un signal numérique compressé qui comporte des données d'en-tête et un corps de signal comprenant notamment des paquets de données, caractérisé en ce que le traitement de la requête comporte une étape de détermination de la position, dans le corps du signal, d'au moins un paquet de données correspondant à la requête en fonction, d'une part, de la longueur des données d'en-tête et, d'autre part, d'au moins un marqueur de pointage présent dans le signal et adapté à fournir la longueur de la partie du corps précédant le paquet de données considéré.

Corrélativement, l'invention vise également un dispositif de traitement d'une requête provenant d'un premier appareil de communication connecté à travers un réseau de communication à un deuxième appareil de communication distant, le dispositif comportant des moyens de réception de la requête d'obtention de données numériques d'un signal numérique compressé qui comporte des données d'en-tête et un corps de signal comprenant notamment des paquets de données, caractérisé en ce que le dispositif comporte, pour le traitement de la requête, des moyens de détermination de la position, dans le corps du signal, d'au moins un paquet de données correspondant à la requête en fonction, d'une part, de la longueur des données d'en-tête et, d'autre part, d'au moins un marqueur de pointage présent dans le signal et adapté à fournir la longueur de la partie du corps précédant le paquet de données considéré.

On utilise un ou plusieurs de ces marqueurs de pointage présents dans le signal selon la structure des données présentes dans la partie du corps précédant le paquet recherché et l'on peut donc déterminer, par calcul, la position dans le signal du paquet correspondant à la requête.

On trouve ainsi le paquet recherché plus rapidement que dans l'art antérieur puisqu'il n'est pas nécessaire de parcourir tout ou partie du corps du

signal. Le premier appareil de communication recevra donc plus rapidement le ou les paquets de données correspondant à sa requête.

De plus, la solution proposée est plus simple dans la mesure où elle ne nécessite pas de décoder les données d'en-tête des différents paquets du corps du signal.

Selon une caractéristique, la détermination de la longueur de la partie du corps du signal précédant le paquet de données considéré comporte une étape préalable de détermination de l'ordre d'apparition dudit paquet de données dans le corps du signal, en fonction de paramètres relatifs à la structure et à l'organisation des données dans le signal.

Selon une caractéristique, le signal numérique compressé est partitionné en un nombre n de zones t_i compressées de façon indépendante, $i = 1$ à n et $n \geq 1$, le corps du signal comprenant pour chaque zone des données d'en-tête de zone et un corps de zone contenant des paquets de données de la zone considérée.

Suivant une autre caractéristique dépendant de la précédente, la longueur de la partie du corps du signal précédant le paquet de données considéré est déterminée à partir de :

- au moins un marqueur de pointage adapté à fournir notamment la longueur du ou des paquets de données précédant le paquet de données considéré dans la zone où se trouve ce paquet,

- la longueur des données d'en-tête de la zone où se trouve le paquet considéré et, lorsqu'une ou plusieurs zones précèdent la zone où se trouve le paquet considéré,

- au moins un marqueur de pointage adapté à fournir notamment la longueur de la ou des zones précédentes.

Ainsi, à partir des données d'en-tête on détermine la position d'un paquet dans le corps du signal sans avoir à effectuer d'opérations complexes.

Une fois la position d'un ou de plusieurs paquets déterminée, le procédé comporte une étape d'extraction et une étape de transmission au premier appareil de communication de ce ou ces paquets.



L'invention permet donc de répondre rapidement et simplement à une requête d'obtention de données numériques provenant du premier appareil de communication.

5 Il convient de noter que la requête d'obtention de données numériques peut soit spécifier au moins un paquet de données du signal, soit spécifier une partie de ce signal.

Dans le cas où la requête d'obtention de données spécifie une partie du signal, alors le procédé selon l'invention mis en œuvre dans le deuxième appareil de communication comporte, postérieurement à la réception de cette
10 requête, une étape d'identification du ou des paquets de données nécessaires à la reconstruction du signal spécifiée.

Selon une autre caractéristique, lorsque le ou les marqueurs de pointage nécessaires à la mise en œuvre de l'invention ne sont pas présents dans le signal numérique compressé, alors le procédé selon l'invention
15 comporte une étape préalable de formation de ce ou ces marqueurs dans le signal.

Par ailleurs, lorsque l'appareil de communication émetteur de la requête, à savoir, par exemple, la machine client, reçoit le ou les paquets de données correspondant à sa requête, il les stocke dans un fichier local en cours
20 de construction.

Cependant, cet appareil ne connaît pas la position du ou des paquets dans le fichier lorsque ce dernier contiendra tous les paquets reçus (s'il y en a plusieurs) par l'appareil au cours de la session de communication.

Dans le cas où le paquet qui est en cours de réception est destiné à
25 être positionné derrière les paquets déjà reçus et stockés préalablement dans le fichier, alors ce paquet est écrit à la fin de ce fichier.

Cependant, si le paquet en cours de réception est destiné à être inséré entre plusieurs paquets déjà stockés dans le fichier en cours de construction, alors le processus de lecture / écriture dans ce fichier devient plus
30 long et plus complexe.

En effet, il convient dans ce cas de ré-écrire une partie importante du fichier en cours de construction et, plus particulièrement, d'effectuer les opérations suivantes :

- 5 - détermination de l'ordre d'apparition du paquet reçu dans le fichier en cours de construction,
- mémorisation de la portion de train binaire située à la suite de la position de destination du paquet reçu,
- écriture du paquet à la position de destination,
- écriture de la portion de train binaire mémorisée.

10 Un tel processus risque de ralentir considérablement le traitement des paquets de données reçus dans l'appareil de communication émetteur de la requête.

 Selon un deuxième aspect de l'invention, il serait par conséquent utile de disposer d'un nouveau procédé et d'un nouveau dispositif qui permettent, à
15 un appareil de communication, de traiter les données reçues par ce dernier de façon plus rapide que dans l'art antérieur.

 La présente invention a ainsi pour objet un procédé de traitement de données numériques compressées reçues par un premier appareil de communication connecté à travers un réseau de communication à un deuxième
20 appareil de communication distant, le procédé comportant une étape de réception d'au moins un paquet de données provenant d'un signal numérique compressé présent dans le deuxième appareil et comportant un corps qui comprend notamment des paquets de données, caractérisé en ce que le procédé comporte les étapes suivantes :

- 25 - détermination d'une position à laquelle doit être inséré ledit au moins un paquet de données dans le corps d'un signal numérique compressé dérivé du signal numérique compressé présent dans le deuxième appareil et qui est apte à contenir tout ou partie du corps de ce signal numérique compressé, le signal dérivé comportant également des données d'en-tête, la détermination de
30 la position étant effectuée en fonction, d'une part, de la longueur des données d'en-tête et, d'autre part, d'au moins un marqueur de pointage préalablement

reçu et inséré dans le signal par le premier appareil et qui est adapté à fournir la longueur de la partie du corps précédant ledit au moins un paquet de données,

- insertion dans le corps du signal dérivé dudit au moins un paquet de données à la position ainsi déterminée.

5 Corrélativement, l'invention vise également un dispositif de traitement de données numériques compressées reçues par un premier appareil de communication connecté à travers un réseau de communication à un deuxième appareil de communication distant, le dispositif comportant des moyens de réception d'au moins un paquet de données provenant d'un signal numérique
10 compressé présent dans le deuxième appareil et comportant un corps qui comprend notamment des paquets de données, caractérisé en ce que le dispositif comporte :

- des moyens de détermination d'une position à laquelle doit être inséré ledit au moins un paquet de données dans le corps d'un signal
15 numérique compressé dérivé du signal numérique compressé présent dans le deuxième appareil et qui est apte à contenir tout ou partie du corps de ce signal numérique compressé, le signal dérivé comportant également des données d'en-tête, la détermination de la position étant effectuée en fonction, d'une part, de la longueur des données d'en-tête et, d'autre part, d'au moins un marqueur
20 de pointage préalablement reçu et inséré dans le signal par le premier appareil et qui est adapté à fournir la longueur de la partie du corps précédant ledit au moins un paquet de données,

- des moyens d'insertion dans le corps du signal dérivé dudit au moins un paquet de données à la position ainsi déterminée.

25 Dans ce deuxième aspect de l'invention, on utilise de la même façon que pour le premier aspect de l'invention énoncé plus haut et dans le même but un ou plusieurs marqueurs de pointage présents dans le signal numérique compressé disponible dans le deuxième appareil et qui ont été reçus par le premier appareil.

30 On détermine ainsi la position exacte à laquelle doivent être insérés le ou les paquets de données compressés reçus dans le corps du signal dérivé,

sans procéder, pour ce faire, à la moindre insertion préalable de données dans ce corps.

5 Ainsi, chaque paquet reçu peut directement être inséré à la bonne position dans le corps du signal dérivé. Cette position correspond à la position qu'occuperait le paquet concerné si tous les paquets du signal disponible dans le deuxième appareil étaient reçus par le premier appareil.

Il convient toutefois de noter que cette position ne correspond pas nécessairement à la position qu'occupait le paquet dans le signal disponible dans le deuxième appareil.

10 Selon une caractéristique, le procédé selon ce deuxième aspect de l'invention, comporte les étapes préalables suivantes :

- réception des données d'en-tête provenant du signal numérique compressé d'origine présent dans le deuxième appareil, les données d'en-tête reçues comprenant au moins un marqueur de pointage adapté à fournir la
15 longueur du corps du signal d'origine,

- à partir des données d'en-tête reçues, formation du signal numérique compressé dérivé qui comprend ainsi, comme données d'en-tête, les données d'en-tête reçues et un corps de signal de longueur égale à celle du corps du signal d'origine, le corps du signal dérivé représentant un espace initialement
20 rempli de données arbitraires et qui est destiné à contenir le ou les paquets de données reçus du deuxième appareil.

On forme ainsi à partir des données d'en-tête reçues un squelette du signal dérivé comprenant, d'une part, comme données d'en-tête, les données d'en-tête reçues du signal d'origine présent dans le deuxième appareil de
25 communication et, d'autre part, un corps qui va être rempli au fur et à mesure de la réception de paquets de données provenant de ce deuxième appareil.

Lorsque le signal numérique compressé est partitionné en zones, la longueur de la partie du corps précédant le paquet de données dont on souhaite déterminer la position est déterminée de façon identique à ce qui a été
30 décrit pour le deuxième appareil de communication selon l'invention.

Là aussi, la détermination de la position d'un paquet de données dans le corps du signal dérivé en cours de formation est réalisée de façon simple et rapide dans le premier appareil de communication selon l'invention.

5 Selon une autre caractéristique, le procédé comporte les étapes suivantes :

- réception de données d'en-tête de zone,
 - détermination d'une position à laquelle doivent être insérées les données d'en-tête de zone reçues dans le corps du signal dérivé, la détermination de la position étant effectuée en fonction de la longueur des données d'en-tête du signal dérivé et, lorsqu'une ou plusieurs zones précèdent les données d'en-tête de zone concernées, également en fonction d'un ou plusieurs marqueurs de pointage préalablement reçus et fournissant respectivement la longueur de la ou des zones précédentes,
 - insertion des données d'en-tête de zone reçues à la position ainsi
- 15 déterminée.

Lorsque le signal est partitionné en zones on détermine ainsi la position des en-têtes de zones dans le signal dérivé en cours de formation et à laquelle les données d'en-tête de zones reçues du deuxième appareil de communication doivent être insérées dans ce signal.

20 Là encore, l'utilisation du ou des marqueurs de pointage permet de déterminer cette position de façon simple et rapide.

Selon une caractéristique, le procédé comporte une phase de transformation du signal dérivé en un signal valide qui comporte les étapes suivantes :

- extraction du signal dérivé des données d'en-tête et des paquets de données reçus,
 - formation des données d'en-tête du signal valide à partir des données d'en-tête extraites du signal dérivé,
 - concaténation des paquets de données extraits du signal dérivé
- 25 dans le corps du signal valide, et
- lorsqu'un ou plusieurs paquets de données présents dans le corps du signal d'origine ne sont pas reçus par le premier appareil, concaténation
- 30

respectivement d'un ou de plusieurs paquets vides dans le corps du signal valide suivant le même ordre d'apparition que celui adopté dans le signal dérivé.

De cette façon, on transforme le signal dérivé en un signal valide qui, par exemple, dans le cas où le signal numérique compressé présent dans le deuxième appareil de communication est conforme au standard JPEG2000, va également se révéler dans le premier appareil, après transformation, en un signal valide au sens du standard JPEG2000.

Ceci est particulièrement avantageux dans la mesure où il sera possible d'utiliser un décodeur JPEG2000 indépendant de l'architecture client-serveur utilisée.

Par ailleurs, dans la mesure où le signal dérivé est distinct du signal valide et peut donc être conservé en mémoire dans le premier appareil de communication, il est possible de continuer à recevoir des paquets de données et à les stocker dans ce signal dérivé lors de sessions de communications futures sans qu'il soit pour cela nécessaire de reformer un autre signal dérivé.

On remarquera que la création de paquets vides dans le corps du signal valide simulant la présence de paquets non reçus permet ainsi de rendre ce dernier valide au sens du standard JPEG2000.

Suivant une autre possibilité, le procédé selon l'invention comporte les étapes suivantes :

- parcours des données contenues dans le corps du signal dérivé,
- lorsque les données parcourues ne correspondent pas à un paquet de données reçu du deuxième appareil, transformation de l'espace rempli par les données concernées en un paquet vide et,
- décalage de façon adaptée des données constituant la suite du corps du signal dérivé.

Selon cette approche, le signal dérivé est transformé en un signal valide sans pour autant créer un autre signal.

On reste ainsi à l'intérieur du signal dérivé qui va se transformer au fur et à mesure des opérations effectuées sur ses données en un signal valide.

De ce fait, les ressources nécessaires au stockage des données seront réduites par rapport au cas précédant où l'on conserve un signal valide et un signal dérivé.

5 L'invention concerne également un appareil de communication comportant un dispositif de traitement d'une requête ou de données reçues, selon l'aspect considéré, tel que brièvement exposé ci-dessus.

Selon un autre aspect, l'invention vise aussi :

10 - un moyen de stockage d'informations lisible par un ordinateur ou un microprocesseur comportant des instructions de code d'un programme d'ordinateur pour l'exécution des étapes du procédé selon l'invention tel que celui exposé brièvement ci-dessus, et

15 - un moyen de stockage d'informations amovible, partiellement ou totalement, lisible par un ordinateur ou un microprocesseur comportant des instructions de code d'un programme d'ordinateur pour l'exécution des étapes du procédé selon l'invention tel que celui brièvement exposé ci-dessus.

20 Selon encore un autre aspect, l'invention vise un programme d'ordinateur chargeable dans un appareil programmable, comportant des séquences d'instructions ou portions de code logiciel pour mettre en œuvre des étapes du procédé de l'invention tel que brièvement exposé ci-dessus, lorsque ledit programme d'ordinateur est chargé et exécuté sur l'appareil programmable.

25 Les caractéristiques et avantages relatifs au dispositif de traitement d'une requête ou de données reçues, à l'appareil de communication comportant un tel dispositif, aux moyens de stockage d'informations et au programme d'ordinateur étant les mêmes que ceux exposés ci-dessus concernant le procédé selon l'invention, ils ne seront pas rappelés ici.

D'autres caractéristiques et avantages de la présente invention apparaîtront plus clairement à la lecture de la description qui va suivre, faite en référence aux dessins annexés, sur lesquels :

30 - la figure 1 représente de manière schématique une architecture de communication de type client-serveur dans laquelle l'invention est mise en œuvre ;

- la figure 2 représente de manière schématique un appareil de communication (machine client ou serveur) dans lequel l'invention est mise en œuvre ;
- la figure 3 représente un algorithme de formation de marqueurs de pointage ;
- la figure 4 est un algorithme détaillant les opérations effectuées à l'étape E4 de l'algorithme de la figure 3 ;
- la figure 5 est un algorithme détaillant les opérations effectuées à l'étape E11 de l'algorithme de la figure 3 ;
- la figure 6 est une représentation schématique de la structure d'un train binaire d'un signal d'image conforme au standard JPEG2000 ;
- la figure 7 est un algorithme de traitement d'une requête mis en œuvre par le deuxième appareil de communication (serveur) ;
- la figure 8 est un algorithme détaillant les opérations effectuées à l'étape E52 de l'algorithme de la figure 7 ;
- la figure 9 est un algorithme détaillant les opérations effectuées à l'étape E54 de l'algorithme de la figure 7 ;
- la figure 10 est un algorithme mis en œuvre par le deuxième appareil de communication (serveur) et qui concerne le traitement d'une requête précisant une zone d'intérêt souhaitée par l'utilisateur ;
- la figure 11 est un algorithme mis en œuvre par le premier appareil de communication (machine client) et qui concerne le traitement de données reçues par cet appareil ;
- la figure 12 est un algorithme détaillant les opérations effectuées à l'étape E102 de l'algorithme de la figure 11 ;
- la figure 13 est une représentation schématique de la structure d'un signal dérivé formé dans le premier appareil de communication selon l'invention ;
- la figure 14 est un algorithme détaillant les opérations effectuées à l'étape E118 de l'algorithme de la figure 11 ;
- la figure 15 est un algorithme illustrant une variante de réalisation de l'algorithme de la figure 11 qui concerne le traitement, par le premier appareil

de communication (machine client), d'une requête visant à obtenir une zone d'intérêt souhaitée par l'utilisateur, ainsi que la réponse à cette requête ;

- la figure 16 est une représentation schématique d'un signal dérivé formé dans le premier appareil de communication et du signal valide issu de ce dernier ;

- la figure 17 est un algorithme de transformation du signal dérivé en un signal valide ;

- la figure 18 représente une variante de réalisation de l'algorithme de la figure 17.

La **figure 1** représente de façon très schématique une architecture de communication de type client-serveur dans laquelle l'invention est avantageusement mise en œuvre.

Comme représenté sur la figure 1, un premier appareil de communication 30 (machine client) est connecté à un deuxième appareil de communication 32 (serveur) à travers un réseau de communication 34.

Dans cette architecture, un signal numérique compressé, par exemple un signal d'image compressé, est stocké dans le serveur 32.

Un utilisateur interagissant avec la machine client 30, par l'intermédiaire d'une interface homme-machine 36, va chercher à obtenir des données numériques provenant du signal compressé stocké au serveur afin de visualiser l'image de façon interactive. Ceci s'accompagne de la formation, dans la machine client 30, d'un signal numérique dérivé du signal numérique compressé du serveur.

Cet échange de données prend la forme de requêtes et de réponses qui sont transmises entre la machine client 30 et le serveur 32 via le réseau de communication 34.

Selon un premier aspect, l'invention est mise en œuvre dans le serveur 32 pour traiter une requête provenant de la machine client 30.

Selon un deuxième aspect, l'invention est mise en œuvre dans la machine client 30 pour traiter, dans celle-ci, des données reçues du serveur 32 et, plus particulièrement, procéder au stockage ("caching") de ces données.

Les données ainsi reçues par la machine client 30 sont stockées dans une mémoire de type cache, ce qui évite d'avoir à transmettre au serveur à plusieurs reprises des requêtes correspondant à une même partie de signal (zone d'intérêt) demandée par l'utilisateur à des moments différents.

5 Il convient de noter que les données reçues par la machine client sont ensuite décompressées et affichées pour pouvoir être utilisées par l'utilisateur via l'interface homme-machine 36.

On notera que pour la description qui va suivre le signal numérique compressé est plus particulièrement un signal d'image bien qu'il puisse
10 également prendre la forme, par exemple, d'un signal vidéo ou d'un signal audio.

Plus particulièrement, le signal numérique compressé correspond à un signal d'image conforme au standard JPEG2000.

Ainsi, les signaux stockés respectivement dans le serveur 32 et dans
15 la machine client 30 sont stockés sous la forme de fichiers informatiques, à savoir un fichier JPEG2000 original pour le serveur et un fichier dit de cache pour la machine client.

On notera par ailleurs que le signal numérique compressé original
20 peut provenir d'autres appareils de communication qui sont connectés au serveur par un autre réseau de communication.

Dans l'exemple considéré, le serveur 32 est, par exemple, un ordinateur et la machine client 30, un micro-ordinateur, voire un appareil photographique numérique, un téléphone portable, un assistant personnel du type PDA (connu en terminologie anglo-saxonne sous le terme "Personal Digital
25 Assistant")....

En outre, le réseau de communication 34 peut être, par exemple, de type filaire ou radio.

En référence à la **figure 2**, est décrit un exemple d'appareil programmable 100 mettant en œuvre l'invention. Cet appareil est adapté à
30 traiter un signal numérique compressé ou des informations et données provenant de ce signal ou d'une requête émanant d'un autre appareil de communication.

Chaque appareil de communication de la figure 1 prend, par exemple, la forme de l'appareil de la figure 2 qui inclut un dispositif selon l'invention, c'est-à-dire possédant tous les moyens nécessaires à la mise en œuvre de l'invention, ou constitue lui-même un tel dispositif selon l'invention.

5 Selon le mode de réalisation représenté à la figure 1, un appareil mettant en œuvre l'invention est par exemple un ordinateur ou un micro-ordinateur 100 connecté à différents périphériques, par exemple une caméra numérique 101 (ou un scanner, ou tout moyen d'acquisition ou de stockage d'image) reliée à une carte graphique et fournissant des données.

10 L'appareil 100 de la figure 2 comporte un bus de communication 102 auquel sont reliés :

- une unité centrale 103 (microprocesseur),
 - une mémoire morte 104, comportant un programme "Progr",
 - une mémoire vive 106, comportant des registres adaptés à
- 15 enregistrer des variables modifiées au cours de l'exécution du programme précité,
- un écran 108 permettant de visualiser les données décompressées (quand il s'agit d'une machine client) ou de servir d'interface avec l'utilisateur qui pourra paramétrer certains modes de fonctionnement du serveur ou de la
- 20 machine client selon le cas, à l'aide d'un clavier 110 ou de tout autre moyen, tel que par exemple une souris,
- un disque dur 112,
 - un lecteur de disquette 114 adapté à recevoir une disquette 116,
 - une interface de communication 118 avec le réseau de
- 25 communication 34 de la figure 1 et qui est apte à traiter des requêtes provenant d'autres appareils de communication et à transmettre des données numériques compressées en réponse à ces requêtes ou à recevoir des données compressées et qui sont à traiter par l'appareil,
- une carte d'entrée/sortie 122 reliée à un microphone 124 (les
- 30 données à traiter selon l'invention constituent alors un signal audio).

Le bus de communication permet la communication entre les différents éléments inclus dans le micro-ordinateur 100 ou reliés à lui. La

représentation du bus n'est pas limitative et, notamment, l'unité centrale est susceptible de communiquer des instructions à tout élément du micro-ordinateur 100 directement ou par l'intermédiaire d'un autre élément du micro-ordinateur 100.

5 Le programme noté "Progr" permettant à l'appareil programmable de mettre en œuvre l'invention, peut être stocké par exemple en mémoire morte 104 (appelée ROM sur le dessin).

Bien qu'un seul programme soit identifié, il est possible d'avoir plusieurs programmes ou sous programmes pour mettre en œuvre l'invention.

10 Selon une variante, la disquette 116, tout comme le disque dur 112, peuvent contenir des données compressées et stockées ainsi que le code de l'invention qui, une fois lu par l'appareil 100, sera stocké dans le disque dur 112.

En seconde variante, le programme pourra être reçu pour être stocké de façon identique à celle décrite précédemment par l'intermédiaire du réseau de communication 34.

15 Les disquettes peuvent être remplacées par tout support d'information tel que, par exemple, un CD-ROM ou une carte mémoire. De manière générale, un moyen de stockage d'information, lisible par un ordinateur ou par un microprocesseur, intégré ou non à l'appareil, éventuellement amovible, mémorise un programme mettant en œuvre le procédé selon l'invention.

De manière plus générale, le programme pourra être chargé dans un des moyens de stockage de l'appareil 100 avant d'être exécuté.

25 L'unité centrale 103 va exécuter les instructions relatives à la mise en œuvre de l'invention, instructions stockées dans la mémoire morte 104 ou dans les autres éléments de stockage. Lors de la mise sous tension, le ou les programmes, qui sont stockés dans une mémoire non volatile, par exemple la mémoire ROM 104, sont transférés dans la mémoire vive RAM 106 qui contiendra alors le code exécutable de l'invention, ainsi que des registres pour mémoriser les variables nécessaires à la mise en œuvre de l'invention.

30 Il convient de noter que l'appareil de communication qui est apte à mettre en œuvre l'invention peut également être un appareil programmé.

Cet appareil contient alors le code du ou des programmes informatiques par exemple figé dans un circuit intégré à application spécifique (ASIC).

La **figure 3** illustre un algorithme comportant différentes instructions ou portions de code logiciel correspondant à des étapes du procédé selon l'invention.

Le programme informatique noté "Progr" qui est basé sur cet algorithme est stocké dans la mémoire morte 104 de la figure 2 et, à l'initialisation du système, est transféré dans la mémoire vive 106.

Le programme est ensuite exécuté par l'unité centrale 103, ce qui permet ainsi de mettre en œuvre le procédé selon l'invention dans l'appareil de la figure 2.

Cet algorithme est mis en œuvre au niveau du deuxième appareil de communication selon l'invention, à savoir le serveur 32 de la figure 1.

Cet algorithme décrit de façon schématique les étapes qui sont mises en œuvre dans ce serveur pour effectuer un transcodage du signal numérique compressé et qui consiste, plus particulièrement, à former des marqueurs de pointage dans le train binaire du signal, à savoir, dans l'exemple de réalisation considéré, le train binaire du signal d'image conforme au standard JPEG2000.

On rappellera que, selon cette norme, un signal numérique d'image compressé comporte des données d'en-tête constituant un en-tête principal, des données d'en-tête de tuiles ou zones t_i suivant lesquelles est partitionné le signal (une tuile représente, de façon compressée, une partie rectangulaire du signal d'image qui comporte toujours au moins une tuile) et, pour chaque tuile, un corps de tuile ou de zone comportant des paquets de données qui sont chacun constitués de données d'en-tête de paquet et d'un corps de paquet.

Le corps de paquet contient à son tour plusieurs blocs de données compressés qui sont représentatives de grandeurs physiques que sont les pixels.

Les données d'en-tête de paquet contiennent notamment une liste de tous les blocs contenus dans le corps de paquet.

Chaque bloc de données compressé est une représentation compressée d'une partie rectangulaire élémentaire du signal d'image qui a été transformé, de manière connue, en sous-bandes de fréquence.

Il convient de noter que chaque bloc de données est compressé sur
5 plusieurs niveaux ou couches de qualité et chaque niveau ou couche de qualité d'un bloc se trouve dans un paquet distinct.

Par ailleurs, les zones ou tuiles t_i précitées sont compressées de façon indépendante.

Chaque paquet de données d'un signal d'image conforme à la norme
10 JPEG2000 contient donc un ensemble de blocs de données compressés correspondant chacun à une tuile, une composante (par exemple : luminance ou chrominance), un niveau de résolution, un niveau de qualité et une position ou localisation spatiale (connu en terminologie anglo-saxonne sous le terme "precinct") donnés.

De retour à l'algorithme de la figure 3, celui-ci débute par une étape
15 E1 de positionnement au début du train binaire du signal d'image considéré.

Au cours de l'étape suivante E2, il est prévu de lire les données d'en-
tête principal du signal et, au cours de l'étape suivante E3, un test est pratiqué
20 afin de déterminer si un marqueur de pointage TLM est présent dans les données d'en-tête principal.

On notera que le marqueur de pointage TLM fournit les longueurs des différentes tuiles qui partitionnent le signal d'image.

Dans l'hypothèse où un tel marqueur n'est pas présent dans le signal
alors l'étape E3 est suivie d'une étape E4 de formation d'un marqueur de
25 pointage TLM dans les données d'en-tête principal du signal.

La formation du marqueur de pointage TLM sera décrite plus en détail ultérieurement, en référence à l'algorithme de la figure 4.

L'étape E4 est suivie d'une étape E5 au cours de laquelle on construit
de nouvelles données d'en-tête principal contenant le marqueur ainsi formé en
30 copiant, dans un nouveau fichier, les données d'en-tête principal du signal d'origine avec le marqueur de pointage TLM ainsi formé.

Au cours de l'étape suivante E6 un test est prévu afin de déterminer si un autre marqueur de pointage noté PLM est présent dans les données d'en-tête principal du signal.

On notera que ce marqueur fournit les longueurs des paquets de données de chacune des zones ou tuiles suivant lesquelles le signal est partitionné.

Lorsqu'un tel marqueur est présent alors l'étape suivante E7 prévoit de copier la partie restante du train binaire constituant le signal d'image dans le nouveau fichier précité, et cette étape est alors suivie d'une étape E8 qui met fin à l'algorithme.

Lorsque le résultat du test pratiqué à l'étape E6 est négatif, alors cette étape est suivie d'une étape E9 au cours de laquelle il est prévu de se positionner à la première tuile du signal, à savoir la tuile d'indice $Tidx = 0$.

L'étape suivante E10 prévoit d'effectuer un test afin de déterminer si un autre marqueur de pointage noté PLT est présent dans les données d'en-tête de la tuile courante $Tidx$.

On notera que ce marqueur de pointage fournit une liste des longueurs des paquets de la tuile considérée.

Si un tel marqueur n'est pas présent dans les données d'en-tête de la tuile ou zone courante concernée, alors l'étape E10 est suivie d'une étape E11 de formation d'un marqueur de pointage PLT.

Cette formation sera détaillée ultérieurement lors de la description faite en référence à la figure 5.

Au cours de l'étape suivante E12 il est prévu de construire de nouvelles données d'en-tête de tuile avec le marqueur de pointage PLT ainsi formé, en effectuant une copie de ces données d'en-tête avec ce marqueur dans le nouveau fichier.

L'étape suivante E13 prévoit d'effectuer une copie des paquets de données de la tuile courante dans le nouveau fichier.

L'étape suivante E14 comporte un test afin de déterminer si la tuile courante est la dernière tuile du signal.

Dans la négative, on passe alors à l'étape E15 au cours de laquelle l'indice de la tuile courante est incrémenté d'une unité et l'étape E10 précédemment décrite est alors de nouveau exécutée.

5 Lorsque l'on a traité la dernière tuile du signal, alors l'étape E14 est suivie d'une étape E16 au cours de laquelle un test est pratiqué afin de déterminer si un nouveau fichier a été créé.

Dans la négative l'étape E16 est directement suivie par l'étape E8 qui met fin à l'algorithme.

10 Dans l'affirmative, l'étape E16 est suivie d'une étape E17 au cours de laquelle un marqueur de fin de train binaire EOC est ajouté à la fin du nouveau train binaire construit comme indiqué précédemment.

Cette étape est alors suivie de l'étape E8 mettant fin à l'algorithme.

15 De retour à l'étape E3, lorsque le résultat du test pratiqué à cette étape est positif, alors on passe à l'étape suivante E18 au cours de laquelle un test est pratiqué afin de déterminer si un marqueur de pointage PLM est présent dans les données d'en-tête principal du signal.

On notera que le marqueur PLM fournit les longueurs des paquets de chaque tuile du signal d'image.

20 Lorsque le résultat du test pratiqué à l'étape E18 est positif, alors aucune modification n'est nécessaire et l'algorithme se termine par l'étape E8.

Lorsque le résultat est négatif, alors on passe à l'étape suivante E9 précédemment décrite au cours de laquelle on se positionne à la première tuile du signal d'image.

25 La boucle décrite ci-dessus et constituée des étapes E9 à E15 permet de parcourir toutes les tuiles du signal d'image afin de déterminer si un marqueur PLT est présent dans leurs données d'en-tête et, dans la négative, à former un tel marqueur.

30 Il convient de noter que l'utilisation d'un marqueur de pointage PLT à la place du marqueur PLM est préférable dans la mesure où il s'ensuit une plus faible utilisation de la bande passante, lors de la transmission progressive de parties d'un signal d'image, dans l'architecture client-serveur représentée à la figure 1.

En effet, dans la mesure où ce sont uniquement les données d'en-tête de tuiles nécessaires à la reconstruction, dans la machine client, de la partie du signal d'image demandée par l'utilisateur qui sont transmises à la machine client, seuls les marqueurs de pointage PLT utiles à reconstruire la partie du signal demandée sont transmis.

Au contraire, dans l'hypothèse où l'on transmet le marqueur de pointage PLM à la machine client, celui-ci étant très volumineux, il s'ensuit que des informations de longueurs de paquets non nécessaires pour décoder et reconstruire la partie du signal demandée par l'utilisateur seront malgré tout transmises.

Comme annoncé plus haut, l'algorithme de la **figure 4** détaille les opérations effectuées à l'étape E4 de l'algorithme de la figure 3.

Cet algorithme comporte différentes instructions ou portions de code logiciel correspondant à des étapes du procédé selon l'invention et fait partie de l'algorithme de la figure 3.

L'algorithme de la figure 4 débute par une étape E20 d'initialisation de l'indice de tuile à 0 afin de se positionner sur la première tuile $Tidx = 0$.

Au cours de l'étape suivante E21 on procède à une recherche dans le train binaire du signal d'image d'un marqueur noté SOT (en terminologie anglo-saxonne "Start of Tile") qui indique le début d'une tuile ou zone de l'image.

Il convient de noter que chaque marqueur SOT contient un champ noté P_{sot} qui fournit la longueur totale de la tuile considérée.

Au cours de l'étape suivante E22, il est prévu d'effectuer une lecture du champ P_{sot} dans le marqueur SOT de début de tuile et d'effectuer une mémorisation de ce champ dans un marqueur de pointage TLM, pour la partie de ce marqueur correspondant à la tuile $Tidx$ considérée.

A cet effet, on notera que le marqueur TLM comporte plusieurs champs TLM (k) correspondant chacun à une valeur de longueur d'une des tuiles d'indice k composant le signal.

Au cours de l'étape suivante E23 un test est pratiqué afin de déterminer si l'indice de la tuile qui vient d'être traitée correspond à l'indice de la dernière tuile.

Dans la négative, l'étape E23 suivie d'une étape E24 au cours de laquelle l'indice de la tuile courante est incrémenté d'une unité, et cette étape est alors suivie de l'étape E21 déjà décrite ci-dessus.

5 L'étape E21, pour les tuiles d'indice supérieur à 0, peut simplement consister à se positionner P_{SOT} octets plus loin que le début de la tuile courante, P_{SOT} désignant un champ du marqueur SOT contenant la longueur en octets de la tuile courante.

10 Lorsque le résultat du test pratiqué à l'étape E23 est positif, alors toutes les tuiles du signal d'image ont été examinées, et l'on passe à l'étape suivante E25 au cours de laquelle un test est pratiqué afin de déterminer si le champ P_{sot} précité pour la dernière tuile examinée est nul.

15 Dans la négative, l'étape E25 est suivie d'une étape E26 au cours de laquelle on effectue un codage du marqueur de pointage TLM qui a été formé pour toutes les tuiles du signal d'image, lors de l'exécution de la boucle constituée des étapes E21 à E24.

Le codage est effectué à l'aide des valeurs du marqueur TLM obtenues et mémorisées pour chacune des tuiles considérées du signal d'image.

20 On notera qu'un tel codage est connu de l'homme de l'art d'après le document "JPEG2000 Part I Final Draft International Standards (corrected and formatted)", ISO/IEC JTC1/SC29 WG1, JPEG2000, September 2000.

L'étape suivante E27 met alors fin à l'algorithme de la figure 4.

25 De retour à l'étape E25, lorsque le résultat du test pratiqué à cette étape est positif, alors on passe à l'étape suivante E28 au cours de laquelle on procède à une recherche du marqueur EOC de fin de train binaire et à un calcul de la longueur de la dernière tuile du signal.

L'étape E28 est ensuite suivie de l'étape E26 déjà décrite ci-dessus.

30 Comme annoncé plus haut, la formation du marqueur de pointage PLT prévue à l'étape E11 de la figure 4 va maintenant être décrite de façon plus détaillée en référence à l'algorithme de la **figure 5**.

Cet algorithme comporte différentes instructions ou portions de codes logiciel correspondant à des étapes du procédé selon l'invention et fait également partie de l'algorithme de la figure 3.

5 L'algorithme de la figure 5 débute par une étape E29 qui prévoit de se positionner juste après les données d'en-tête de la tuile courante.

Cette étape est suivie d'une étape E30 d'initialisation de l'indice de paquet à un premier paquet d'indice $Pidx = 0$.

10 Au cours de l'étape suivante E31 on effectue une initialisation de la position courante du paquet considéré dans le train binaire, cette position $curPos$ étant initialisée à 0.

Au cours de l'étape suivante E32 un test est pratiqué afin de déterminer si un marqueur SOP est présent dans le corps de la tuile considérée.

15 Il convient de noter que ce marqueur SOP (en terminologie anglo-saxonne "Start of Packet") fournit l'indication du début de chaque paquet dans le corps de tuile considéré.

20 On notera que, de manière générale, cet algorithme vise à rechercher les marqueurs SOP successifs présents dans le corps de la tuile considérée, la différence de position dans le train binaire entre deux marqueurs SOP successifs fournissant la longueur du paquet débutant au premier de ces deux marqueurs.

25 De retour à l'étape E32, lorsque le marqueur SOP n'est pas présent dans la tuile considérée, alors l'étape suivante E33 prévoit d'effectuer un décodage de l'entête du paquet courant examiné et ainsi de se positionner au début du corps de ce paquet.

Le décodage de l'en-tête fournit ainsi les longueurs des contributions de chaque bloc de données du paquet au corps de ce dernier.

30 Au cours de l'étape suivante E34, on effectue un calcul de la longueur du corps du paquet courant en additionnant les longueurs des contributions de chaque bloc de données du paquet.

L'étape suivante E35 prévoit de se positionner à la fin du paquet courant, ce qui correspond à la position Pos .

Au cours de l'étape suivante E36 on effectue le calcul de la longueur du paquet total Long [Pidx] en effectuant la différence entre la position Pos courante dans le train binaire et la position curPos correspondant au début du paquet courant.

- 5 Au cours de ces étapes on effectue également une mémorisation de la longueur du paquet considéré.

Au cours de l'étape suivante E37 il est prévu de mettre à jour la position courante curPos qui devient celle correspondant à la position Pos de fin du paquet courant.

- 10 Au cours de l'étape suivante E38 un test est pratiqué afin de déterminer si l'indice courant Pidx correspond à l'indice du dernier paquet.

Dans la négative, cette étape est suivie d'une étape E39 d'incrément d'une unité de l'indice du paquet courant.

L'étape E39 est ensuite suivie de l'étape E33 déjà décrite ci-dessus.

- 15 Dans l'affirmative, l'étape E38 est suivie d'une étape E40 où un codage d'un marqueur de pointage noté PLT est effectué à l'aide des valeurs de longueurs de paquets obtenues et mémorisées à l'étape E36 précitée.

De façon analogue à ce qui était décrit en référence à l'étape E26 de l'algorithme de la figure 4, un tel codage est connu de l'homme de l'art.

- 20 L'étape E40 est ensuite suivie d'une étape E41 mettant fin à l'algorithme de la figure 5.

- De retour à l'étape E32, lorsque le résultat du test pratiqué à cette étape est positif, alors on passe à l'étape suivante E42 qui prévoit de rechercher le premier marqueur SOP présent dans la tuile considérée, ce
- 25 marqueur fournissant ainsi la position courante de référence curPos.

Au cours de l'étape suivante E43 il est prévu de rechercher le marqueur suivant SOP et d'identifier sa position qui est alors notée Pos.

- L'étape suivante E44 prévoit, d'une part, de déterminer la longueur du paquet placé entre les deux marqueurs SOP respectivement identifiés aux
- 30 étapes E42 et E43 en effectuant la différence $Pos - curPos$ et, d'autre part, de mémoriser cette longueur de paquet Long [Pidx].

Au cours de l'étape suivante E45 il est prévu de mettre à jour la position courante dans le train binaire qui devient alors celle du marqueur identifiée à l'étape E43, à savoir Pos.

5 L'algorithme de la figure 5 comporte ensuite un test E46 afin de savoir si l'indice courant du paquet Pidx correspond à l'indice du dernier paquet de la tuile considérée.

Dans la négative, cette étape est suivie d'une étape E47 d'incrément de l'indice du paquet d'une unité.

L'étape E47 est ensuite suivie de l'étape E43 déjà décrite ci-dessus.

10 Lorsque le résultat du test pratiqué à l'étape E46 est positif, alors cette étape est suivie de l'étape E40 de codage d'un marqueur PLT pour la tuile considérée à partir des valeurs de longueurs obtenues et mémorisées à l'étape E44.

15 L'étape E40 déjà décrite plus haut est ensuite suivie de l'étape E41 mettant fin à l'algorithme.

La **figure 6** illustre de façon schématique la structure d'un train binaire d'un signal d'image conforme au standard JPEG2000, le signal d'image étant partitionné en quatre tuiles ou zones 0, 1, 2 et 3.

20 Sur cette figure on a également illustré de façon très schématique le mécanisme de détermination de la position d'un paquet de données particulier dans le train binaire.

La description de ce mécanisme sera effectuée ultérieurement.

25 Ainsi, la figure 6 fait apparaître des données d'en-tête de signal 150 qui sont constituées d'un préambule 152 contenant des métadonnées, d'un en-tête principal MHD référencé 154 et d'en-têtes de tuiles 156, 158, 160 et 162 correspondant respectivement aux données d'en-tête des tuiles 0, 1, 2 et 3.

Chaque en-tête de tuile comprend un marqueur SOT indiquant le début de la tuile considérée, des données T_i , où i désigne l'indice de la tuile considérée et un marqueur SOD indiquant la fin de l'en-tête de tuile.

30 On notera que les données T_i contiennent des marqueurs indiquant des paramètres de codage et d'organisation du train binaire pour la tuile d'indice i .

Par ailleurs, comme représenté sur la figure 6, chaque tuile comporte, non seulement des données d'en-tête de tuile comme on vient de le voir, mais également un corps de tuile référencé 164 pour la tuile 0, 166 pour la tuile 1, 168 pour la tuile 2 et 170 pour la tuile 3.

5 Ces corps de tuile contiennent les paquets de données de la tuile considérée qui sont eux-mêmes constitués de blocs de données comme expliqué plus haut.

Enfin, un marqueur EOC de fin de train binaire ("End of Codestream" en terminologie anglo-saxonne) référencé 172 indique la fin du signal d'image.

10 La **figure 7** est un algorithme comportant différentes instructions ou portions de code correspondant à des étapes du procédé de traitement d'une requête selon l'invention.

Cet algorithme est mis en œuvre dans le deuxième appareil de communication (serveur) représenté à la figure 1.

15 Le programme d'ordinateur "Progr" stocké dans l'appareil de la figure 2 est également basé sur cet algorithme pour son exécution.

L'algorithme de la figure 7 débute par une étape E50 d'attente de réception d'une requête RQ provenant du premier appareil de communication (machine client) représenté à la figure 1.

20 Après réception d'une requête, l'étape E50 est suivie d'une étape E51 au cours de laquelle un test est pratiqué afin de déterminer le type de la requête et, plus particulièrement, s'il s'agit d'une requête visant à obtenir des données d'en-tête de tuile.

25 Dans l'affirmative, l'étape E51 est suivie d'une étape E52 qui prévoit le traitement de la requête concernant les données d'en-tête de tuile, ce qui active l'exécution de l'algorithme de la figure 8 comme indiqué ci-après.

30 Comme représenté sur la **figure 8**, l'algorithme comporte différentes instructions ou portions de code logiciel correspondant à des étapes du procédé de traitement d'une requête selon l'invention et fait partie de l'algorithme de la figure 7.

De manière générale, l'algorithme de la figure 8 concerne la détermination de la position des données d'en-tête de tuile demandées dans la

requête, ainsi que leur extraction du train binaire du signal d'image considéré et leur transmission à l'appareil de communication (machine client) dont provient la requête.

L'algorithme de la figure 8 débute par une étape E61 au cours de laquelle on détermine la position, dans le train binaire du signal d'image, du début des données d'en-tête de tuile requises.

Pour ce faire, on utilise le marqueur de pointage TLM qui est présent dans le signal, comme expliqué lors de la description faite en référence aux figures 3 et 4 précitées.

En effet, la lecture des données d'en-tête principal, notées 154 sur la figure 6, fournit la longueur de ces données d'en-tête, ainsi que des informations contenues dans le marqueur de pointage TLM, à savoir les longueurs des différentes tuiles du signal, la valeur contenue dans le champ TLM (k) fournissant la longueur de la tuile d'indice k.

De ce fait, la position de la tuile recherchée dans le signal est fournie par l'expression suivante :

$$\text{Length (preamble + MHD)} + \sum_{k=0}^{t-1} \text{TLM}(k), \text{ où } k \text{ est l'indice de la tuile}$$

considérée.

Ainsi, reprenant l'exemple de la figure 6, si l'on recherche la tuile 2, l'expression ci-dessus fournit la longueur des données d'en-tête principal 154 incluant le préambule 152, ainsi que la longueur des tuiles 0 et 1.

L'étape suivante E62 prévoit de lire les données d'en-tête de la tuile t considérée, à savoir les données d'en-tête 160 dans l'exemple mentionné ci-dessus, jusqu'au marqueur SOD indiquant la fin des données d'en-tête de tuile.

On procède également, lors de l'étape E62, à l'extraction de la suite d'octets correspondant aux données d'en-tête de tuile recherchées.

L'étape suivante E63 prévoit de transmettre la suite d'octets ainsi lus à l'étape précédente au premier appareil de communication (machine client) représenté sur la figure 1.

Il convient de noter que les données d'en-tête de tuile ainsi identifiées et extraites du train binaire peuvent être transmises marqueur par marqueur et non en une seule fois.

Il est alors mis fin à l'algorithme de la figure 8 par l'étape E64 et l'étape
5 E52 de l'algorithme de la figure 7 est alors suivie de l'étape E50 déjà décrite.

On notera également que, lors de la lecture des données d'en-tête de tuile à l'étape E62, les valeurs des longueurs de paquets qui sont contenues dans le marqueur de pointage PLT mentionné lors de la description faite en référence aux figures 3 et 5 sont également lues et seront utilisées
10 ultérieurement lors de la description faite en référence à l'algorithme de la figure 9.

De retour à l'algorithme de la figure 7, lorsque le résultat du test pratiqué à l'étape E51 est négatif (la requête ne concerne pas des données d'en-tête de tuile), alors cette étape est suivie d'une étape E53 qui vise à
15 poursuivre l'analyse de la requête.

Au cours de cette étape un test est pratiqué afin de déterminer si la requête vise un ou plusieurs paquets de données du signal.

Dans l'affirmative, cette étape est suivie d'une étape E54 visant à traiter la requête de paquets, ce qui active l'exécution de l'algorithme de la
20 figure 9.

L'algorithme de la **figure 9** qui fait partie de l'algorithme de la figure 7 détaille les opérations de traitement d'une requête de paquets comme indiqué ci-après.

Cet algorithme comporte différentes instructions ou portions de code logiciel correspondant à des étapes du procédé de traitement d'une requête
25 selon l'invention.

L'algorithme de la figure 9 vise plus particulièrement à traiter une requête concernant un ensemble de paquets $(t_i, c_i, r_i, p_i, l_i)$ où $i = 0$ à $nb_{\text{paquets}} - 1$.
1.

30 Cet algorithme débute par une étape E70 d'initialisation de l'indice de paquet i à la valeur 0.

Il convient de noter que les notations t , c , r , p et l correspondent respectivement aux numéros de tuile, de composante, de niveau de résolution, de position spatiale ("precinct") et de couche de qualité associés aux paquets de données considérés.

5 Au cours de l'étape suivante E71 on détermine préalablement l'ordre d'apparition du paquet de données concerné dans le corps de la tuile t_i considéré, en fonction de paramètres relatifs à la structure et à l'organisation des données dans le signal.

10 Plus particulièrement, ces paramètres sont fournis par les données d'en-tête du signal.

Le numéro de séquence du paquet considéré qui est déterminé selon l'ordre de progression des données dans le train binaire constituant le signal d'image est fourni par une fonction mathématique $f_{\text{Prog-order}}$ qui dépend des paramètres du paquet et de l'ordre de progression.

15 Pour la suite de l'exposé, on utilisera les notations suivantes :

- $\text{numPrec}(c,r)$ correspond au nombre de positions spatiales contenues dans le niveau de résolution r de la composante c ;

- n_C correspond au nombre de composantes du signal d'image ;

- n_L correspond au nombre de couches de qualité dans la tuile

20 considérée.

Ainsi, l'ordre d'apparition du paquet dans le corps du signal, c'est-à-dire le numéro de ce paquet est fourni par les expressions suivantes qui dépendent de l'ordre de progression considéré dans le signal :

- Ordre Résolution-Couche-Composante-Position :

25

$$f_{RLCP}(c,r,p,l) = \sum_{R=0}^{r-1} \sum_{L=0}^{n_L-1} \sum_{C=0}^{n_C-1} \text{numPrec}(C,R) + \sum_{L=0}^{l-1} \sum_{C=0}^{n_C-1} \text{numPrec}(C,r) + \sum_{C=0}^{c-1} \text{numPrec}(C,r) + p$$

- Ordre Couche-Résolution-Composante-Position :

$$f_{LRCP}(c,r,p,l) = \sum_{L=0}^{l-1} \sum_{R=0}^{n_R-1} \sum_{C=0}^{n_C-1} \text{numPrec}(C,R) + \sum_{R=0}^{r-1} \sum_{C=0}^{n_C-1} \text{numPrec}(C,R) + \sum_{C=0}^{c-1} \text{numPrec}(C,r) + p$$

- Ordre Résolution-Position-Composante-Couche : pour calculer le numéro de séquence recherché pour cet ordre de progression, il convient de déterminer tout d'abord les coordonnées (x,y) de la position spatiale p considérée dans la grille de référence, ce qui s'effectue de manière connue comme suit :

$$x = \begin{cases} tx_0 & \text{si } p \bmod \text{numPrecWide}(c,r) = 0 \\ XRSIZ(c).2^{N_{DL}-r}.2^{PP_x(r,c)} \times k_x & \text{sinon} \end{cases}$$

$$y = \begin{cases} ty_0 & \text{si } p \div \text{numPrecWide}(c,r) = 0 \\ YRSIZ(c).2^{N_{DL}-r}.2^{PP_y(r,c)} \times k_y & \text{sinon} \end{cases}$$

$$\text{où : } \begin{cases} k_x = p \bmod \text{numPrecWide}(c,r) + \lfloor tx_0 / 2^{PP_x(r,c)} \rfloor \\ k_y = p \div \text{numPrecWide}(c,r) + \lfloor ty_0 / 2^{PP_y(r,c)} \rfloor \end{cases}$$

Dans les expressions ci-dessus, les grandeurs numPrecWide(c,r), XRSIZ, YRSIZ, tx0, ty0, trx0, try0, PPx et PPy sont connues de l'homme du métier et sont définies dans le document "JPEG2000 Part I Final Draft International Standards (corrected and formatted)", ISO/IEC JTC1/SC29 WG1, JPEG2000, September 2000. La grandeur N_{DL} représente le nombre de niveaux de décompositions en sous-bandes dans la composante considérée.

Une fois ces grandeurs calculées, le numéro de séquence recherché est donné par la formule suivante :

$$f_{RPCL}(c,r,p,l) = \sum_{R=0}^{r-1} \sum_{C=0}^{nC-1} \text{numPrec}(C,R).nL + \sum_{C=0}^{nC-1} \text{nPrecBefore}(r,C,x-1,y).nL + \sum_{C=0}^{c-1} \text{precExists}(res,C,x,y).nL + l$$

où precExists(r,c,x,y) est égal à 1 s'il existe une position spatiale aux coordonnées (x,y) dans la composante c et le niveau de résolution r, et est égal à 0 sinon. Par ailleurs, la fonction nPrecBefore(r,c,x-1,y) représente le nombre de positions spatiales précédant la position (x-1,y), incluse, dans la composante c et le niveau de résolution r. Elle est donnée par l'expression suivante (fournie par le standard JPEG2000):

$$nPrecBefore(r, c, x-1, y) = \left\lfloor \frac{\left\lceil \frac{x-1}{XRSIZ(c).2^{N_{DL}-r}} \right\rceil}{2^{PP_x(r, c)}} \right\rfloor - \left\lfloor \frac{trx_0}{2^{PP_x(r, c)}} \right\rfloor +$$

$$numPrecWide(r, c) \cdot \left(\left\lfloor \frac{\left\lceil \frac{y}{YRSIZ(c).2^{N_{DL}-r}} \right\rceil}{2^{PP_y(r, c)}} \right\rfloor - \left\lfloor \frac{try_0}{2^{PP_y(r, c)}} \right\rfloor \right)$$

- **Ordre Position-Composante-Résolution-Couche** : de la même manière que pour le précédent ordre de progression, le numéro de séquence est donné par l'expression suivante :

5

$$f_{PCRL}(c, r, p, l) = \sum_{C=0}^{nC-1} \sum_{R=0}^{nR-1} nPrecBefore(C, R, x-1, y).nL +$$

$$\sum_{C=0}^{c-1} \sum_{R=0}^{r-1} precExists(R, C, x, y).nL + l$$

- **Ordre Composante-Position-Résolution-Couche** :

$$f_{CPRL}(c, r, p, l) = \sum_{C=0}^{c-1} \sum_{R=0}^{nR-1} numPrec(C, R).nL + \sum_{R=0}^{nR-1} nPrecBefore(c, R, x-1, y).nL +$$

$$\sum_{R=0}^{r-1} precExists(c, R, x, y).nL + l$$

Il convient de noter que le calcul de la grandeur numPrec(c,r) est connu de l'homme de l'art et est fourni par la partie I du standard JPEG2000 dont les références ont été données plus haut.

10

Le numéro de l'ordre d'apparition (numéro de séquence) du paquet d'indice i dans la tuile considérée est ainsi fourni par l'expression $SN_i = f_{Prog-order}(c_i, r_i, p_i, l_i)$.

15

L'étape E71 est ensuite suivie d'une étape E72 au cours de laquelle le traitement de la requête prévoit de déterminer la position du paquet de données considéré dans le corps du signal en fonction, d'une part, de la longueur des données d'en-tête et, d'autre part, d'au moins un marqueur de pointage présent dans le signal et qui est adapté à fournir la longueur de la partie du corps qui précède le paquet considéré.

20

Plus particulièrement, cette position est déterminée en fonction de :

- la longueur des données d'en-tête principal du signal,
 - les longueurs des zones ou tuiles suivant lesquelles le signal est partitionné et qui précèdent la tuile où se trouve le paquet recherché,
 - la longueur des données d'en-tête de la zone ou tuile courante
- 5 contenant le paquet recherché,
- ainsi que la longueur des paquets de données qui précèdent le paquet de données recherché dans la zone ou tuile où se trouve ce paquet.

Plus particulièrement, la longueur des zones ou tuiles précédant celle d'indice t où se trouve le paquet recherché est fournie par la partie

10 correspondante des données constituant le marqueur de pointage TLM et qui sont notées $TLM(k)$, où $k = 0$ à $t-1$.

Par ailleurs, la longueur des paquets de données précédant le paquet de données considéré dans la zone ou tuile où se trouve ce paquet est fournie par les valeurs des longueurs des paquets de la tuile présentes dans le

15 marqueur de pointage PLT de la tuile considérée.

Ainsi, l'expression de la position du paquet de numéro de séquence SN_i (ordre d'apparition dans le corps du signal) est fournie par l'expression suivante :

$$\text{Position}(t_i, SN_i) = \text{Length}(\text{preamble} + \text{MHD}) + \sum_{k=0}^{t_i-1} TLM(k) + \text{Length}(\text{THD}_{t_i}) + \sum_{l=0}^{SN_i-1} PLT_{t_i}(SN_i),$$

où $\text{Length}(\text{THD}_{t_i})$ représente la longueur des données d'en-tête de la zone ou tuile courante t_i .

20 La figure 6 illustre le mécanisme de détermination de la position du paquet recherché, celui-ci étant représenté dans le corps de la tuile 1.

On a également représenté sur cette figure la longueur $PLT_{t_i}(SN_i)$ du paquet recherché dans une tuile du signal.

On voit ainsi que l'utilisation des marqueurs de pointage TLM et PLT

25 fournit un moyen simple pour déterminer la position du paquet recherché dans le corps du signal sans avoir à parcourir tout ou partie du train binaire du signal.

Par ailleurs, l'utilisation de ces marqueurs permet de ne plus avoir à décoder les données d'en-tête des paquets du train binaire.

L'invention permet ainsi de simplifier considérablement une tâche qui se révélait auparavant relativement complexe.

L'utilisation des marqueurs de pointage TLM et PLT permet également d'aboutir plus rapidement qu'auparavant à la détermination de la position du ou
5 des paquets de données recherchés.

L'algorithme de la figure 9 comporte ensuite une étape E73 d'extraction d'une suite d'octets de longueur PLT_{t_i} (SN_i) correspondant à la longueur du paquet dont la position vient d'être déterminée à l'étape E72 et qui est représentée sur la figure 6.

10 L'étape suivante E74 prévoit de transmettre au premier appareil de communication (machine client) représenté à la figure 1, la suite d'octets extraite à l'étape précédente et correspondant à un des paquets de données spécifiés dans la requête.

On notera que cette réponse consécutive à la requête provenant du
15 premier appareil de communication est effectuée au format du protocole utilisé pour la transmission de données.

L'étape 74 est alors suivie d'une étape E75 qui prévoit d'effectuer un test sur l'indice de paquet, afin de déterminer si tous les paquets visés dans la requête ont été trouvés, extraits et transmis au premier appareil de
20 communication.

Dans la négative, cette étape est suivie d'une étape E76 d'incrémentatation de l'indice de paquet i d'une unité, afin de traiter le paquet suivant spécifié dans la requête, et l'on passe alors à l'étape E71 déjà décrite ci-dessus.

25 Lorsque l'indice de paquet i correspond à la valeur $nb_{\text{paquets}} - 1$, alors il est mis fin à l'algorithme par une étape E77.

De retour à la figure 7, la fin de l'algorithme de la figure 9 marque la fin de l'exécution de l'étape E54, et celle-ci est suivie de l'étape E50 précédemment décrite.

30 Au cours de cette étape, le deuxième appareil de communication (serveur) est en attente de réception d'une prochaine requête RQ en provenance du premier appareil de communication (machine client).

De retour à l'étape E53 de la figure 7, lorsque le résultat du test pratiqué à cette étape est négatif, alors un test supplémentaire est pratiqué à l'étape suivante E55.

5 Au cours de cette étape, le test pratiqué vise à savoir si la requête reçue du premier appareil de communication correspond à un signal de fin de communication.

10 Dans la négative, l'étape E55 est suivie d'une étape E56 au cours de laquelle il est prévu d'émettre un signal d'erreur à destination du premier appareil de communication, afin de l'informer que la requête reçue est de type inconnu.

L'étape E56 est alors suivie de l'étape E50 précédemment décrite.

Lorsque le résultat du test pratiqué à l'étape E55 est positif, c'est-à-dire que la requête vise un signal de fin de communication, alors il est mis fin à l'algorithme de la figure 7 par l'étape E57.

15 L'algorithme de la **figure 10** concerne un algorithme de traitement d'une requête reçue par le deuxième appareil de communication (serveur) de la figure 1, dans le cas où cette requête spécifie une partie du signal d'image présent dans cet appareil.

20 Plus particulièrement, cette partie de signal décrit une zone d'intérêt de l'image qui est souhaitée par l'utilisateur au niveau du premier appareil de communication (machine client) de la figure 1.

L'algorithme de la figure 10 comporte différentes instructions ou portions de code correspondant à des étapes du procédé de traitement d'une requête selon l'invention.

25 Le programme d'ordinateur "Progr" stocké dans l'appareil de la figure 2 est également basé sur l'algorithme de la figure 10.

30 Cet algorithme débute par une étape E80 de réception d'une requête provenant du premier appareil de communication et qui spécifie les positions (x, y) et les dimensions (w, h) de la partie du signal souhaités, ainsi que le niveau de résolution et le niveau de qualité de reconstruction demandés par l'utilisateur.

L'étape E80 est suivie d'une étape E81 au cours de laquelle le deuxième appareil de communication détermine les tuiles ou zones du signal d'image qui sont nécessaires pour satisfaire à la requête reçue.

On notera par exemple l'ensemble des tuiles nécessaires à la
5 satisfaction de cette requête : T_0, \dots, T_N .

L'étape suivante E82 prévoit d'initialiser l'indice de tuile t à la valeur T_0 .

L'étape suivante E83 fait appel à l'algorithme de la figure 8 pour
transmettre au premier appareil de communication (machine client) les données
10 d'en-tête de tuile correspondant à la tuile courante.

Une fois ces données d'en-tête de tuile transmises, le deuxième
appareil de communication doit déterminer, pour la tuile considérée, les
paquets de données du train binaire du signal d'image qui sont nécessaires à la
reconstruction de la partie du signal visée dans la requête.

15 Ceci nécessite une projection de la partie du signal considérée dans
les sous-bandes de fréquence de l'image.

Dès lors que la détermination des paquets nécessaires à la
reconstruction de la partie du signal requise par le client dans la tuile t
considérée a été effectuée, alors l'étape E84 est suivie de l'étape E85.

20 Cette étape vise à extraire et à transmettre au premier appareil de
communication (machine client) les paquets de données correspondant à la
requête et qui ont été identifiés à l'étape E84.

On notera que pour la mise en œuvre de l'étape E85 il est nécessaire
de faire appel à l'algorithme de la figure 9 qui prévoit de déterminer la position
25 de chaque paquet dans le train binaire, son extraction ainsi que sa
transmission.

L'étape E85 est alors suivie d'une étape E86 au cours de laquelle un
test est pratiqué afin de déterminer si la tuile nécessaire considérée d'indice T
est la dernière tuile de l'ensemble des tuiles nécessaires pour satisfaire à la
30 requête client.

Dans la négative, l'étape E87 est effectuée.

Au cours de cette étape il est prévu d'incrémenter d'une unité la valeur d'indice T afin de traiter, dans l'algorithme de la figure 10, la prochaine tuile nécessaire (étape E83) pour satisfaire à la requête du client.

Lorsque le résultat du test pratiqué à l'étape E86 est positif, alors
5 toutes les tuiles nécessaires pour satisfaire à la requête du client ont été traitées et les paquets correspondants transmis au premier appareil de communication (machine client).

Il est alors mis fin à l'algorithme de la figure 10 par l'étape E88.

L'algorithme de la **figure 11** concerne le traitement, par le premier
10 appareil de communication (machine client) représenté sur la figure 1, de données reçues par cet appareil, lorsque les requêtes qu'il a transmises au deuxième appareil de communication sont des requêtes visant à obtenir un ou plusieurs paquets de données du signal.

Le mode de fonctionnement du premier appareil de communication
15 qui est illustré par l'algorithme de la figure 11 est complémentaire du mode de fonctionnement du deuxième appareil de communication (premier aspect de l'invention) présenté lors de la description des algorithmes des figures 7 à 9.

Cet algorithme et les suivants illustrent le second aspect de l'invention.

L'algorithme de la figure 11 comporte différentes instructions ou
20 portions de code correspondant à des étapes du procédé de traitement de données reçues par le premier appareil de communication selon l'invention.

Dans le cas où l'appareil de communication représenté à la figure 2 correspond au premier appareil de communication (machine client) de la figure 1, alors le programme d'ordinateur "Progr" stocké dans l'appareil de la figure 2
25 est basé sur l'algorithme de la figure 11, ainsi que les algorithmes suivants.

On notera que l'algorithme de la figure 11 illustre les opérations effectuées par le premier appareil de communication et les relations entre ces opérations afin de rapatrier les données compressées nécessaires à la reconstruction de la partie de signal requise par l'utilisateur, et afin de stocker
30 celles-ci dans un fichier appelé fichier de cache.

L'algorithme de la figure 11 débute par une première étape E100 au cours de laquelle un test est pratiqué afin de déterminer si les données d'en-

tête principal du signal stocké dans le deuxième appareil de communication ont déjà été reçues par le premier appareil et si le fichier de cache a déjà été créé.

Dans la négative, l'étape E100 est suivie d'une étape E101 au cours de laquelle le premier appareil de communication transmet au deuxième
5 appareil une requête visant à obtenir les données d'en-tête principal (incluant le préambule) du signal d'image.

Cette étape est suivie d'une étape E102 au cours de laquelle il est prévu de créer un signal numérique compressé dérivé du signal numérique compressé présent dans le deuxième appareil sous la forme d'un fichier de
10 cache.

Les opérations effectuées lors de cette étape 102 sont illustrées à la **figure 12** qui représente l'algorithme de création d'une version préliminaire du fichier de cache.

Cet algorithme qui fait également partie de l'algorithme de la figure 11
15 débute par une première étape E103 au cours de laquelle les données d'en-tête principal reçues du deuxième appareil de communication sont lues.

Ceci permet d'obtenir au moins un marqueur de pointage adapté à fournir la longueur du corps du signal numérique compressé d'origine.

Plus particulièrement, cette information est fournie par les longueurs
20 des tuiles TLM (k) où $k = 0, \dots, \text{nb_tuiles} - 1$, ces informations étant contenues dans le marqueur de pointage TLM des données d'en-tête principal.

Au cours de l'étape suivante E104, il est prévu de former un signal numérique compressé dérivé à partir des données d'en-tête reçues et qui comprend ainsi, comme données d'en-tête, les données d'en-tête qui viennent
25 d'être reçues.

Plus particulièrement, ces données d'en-tête constituées du préambule et des données d'en-tête principal sont écrites dans un nouveau fichier qui constituera le fichier de cache déjà mentionné.

L'algorithme de la figure 12 comporte ensuite une étape E105 au
30 cours de laquelle il est prévu de remplir le corps du signal dérivé de données arbitraires, de manière à constituer un espace de même taille que le corps du signal d'origine présent dans le deuxième appareil de communication.

Le corps du signal dérivé sera ainsi apte à recevoir tout ou partie du corps du signal numérique compressé d'origine présent dans le deuxième appareil, à savoir les données d'en-tête de zones ou tuiles et les paquets qui seront transmis pendant la session de communication.

- 5 Plus particulièrement, on procède, au cours de l'étape E105, à l'écriture dans les tuiles 0 à nb_tuiles-1 d'un nombre d'octets arbitraire égal à la

$$\text{somme} \sum_{k=0}^{\text{nb_tuiles}-1} \text{TLM}(k)$$

Il est ensuite mis fin à l'algorithme de la figure 12 par l'étape E106.

- 10 La figure 13 illustre la structure du signal dérivé formé dans le premier appareil de communication et stocké dans le fichier de cache.

Comme représenté sur cette figure, les données d'en-tête du signal dérivé sont constituées du préambule et des données d'en-tête principal respectivement notés 152 et 154, les données correspondant aux données d'en-tête portant les mêmes références sur la figure 6.

- 15 Sur la figure 13, on a également représenté dans le corps du signal dérivé l'espace rempli de données arbitraires et qui est réservé aux données d'en-tête de tuiles et aux données constitutives des corps de tuiles destinées à être reçues du deuxième appareil de communication.

- 20 On notera ainsi que l'espace 180 est réservé pour recevoir les données de la tuile 0, l'espace 182 pour recevoir les données de la tuile 1, l'espace 184 pour les données de la tuile 2 et l'espace 186 pour les données de la tuile 3.

Le marqueur de fin de signal 172 de la figure 6 étant indiqué également en fin du signal dérivé sur la figure 13.

- 25 De retour à l'algorithme de la figure 11, on considère la situation où l'utilisateur a émis une requête en vue de l'obtention d'une partie du signal sous la forme, par exemple, d'une position et d'une dimension d'une sous-image de ce signal, ainsi que du niveau de résolution et du niveau de qualité souhaités.

- 30 On remarquera que cette requête de l'utilisateur est interprétée par le premier appareil de communication (machine client) qui identifie alors les

données du signal correspondant à cette requête. L'appareil formule ensuite une requête spécifiant les données voulues, à destination du deuxième appareil de communication.

Il convient toutefois de noter que l'hypothèse où la requête transmise
5 par le premier appareil de communication visait directement une partie de ce signal (position, dimension,...) a été traitée lors de la description faite en référence à l'algorithme de la figure 10.

Le fonctionnement du premier appareil de communication émetteur d'une telle requête sera décrit ultérieurement en référence à la figure 15.

10 Pour l'instant, dans le contexte de l'algorithme de la figure 11, le premier appareil de communication interprète la requête de l'utilisateur spécifiant une partie du signal et procède de la façon suivante.

Au cours d'une étape E112 il est prévu de déterminer l'ensemble des
15 tuiles ou zones du signal qui sont concernées par la partie de signal demandée par l'utilisateur et qui sont donc nécessaires pour satisfaire à la requête.

Cet ensemble de tuiles est noté T_0, \dots, T_N .

Au cours de l'étape suivante E113 on initialise l'indice de tuile t à la valeur T_0 .

L'étape E113 est suivie d'une étape E114 de transmission au
20 deuxième appareil de communication d'une requête visant à obtenir les données d'en-tête de la tuile t considérée.

Dans le déroulement normal des opérations, entre cette étape et l'étape suivante E115 le deuxième appareil de communication représenté à la figure 1 exécute les algorithmes des figures 7 et 8 afin de traiter la requête
25 reçue du premier appareil de communication.

Au cours de l'étape E115, le premier appareil de communication procède à la réception des données d'en-tête de tuiles précédemment demandées et à leur insertion dans le corps du signal dérivé, à la position de destination déterminée lors de l'étape E113.

30 On notera que pour mettre en œuvre cette étape de détermination de la position de destination des données d'en-tête de la tuile t considérée on fait appel à l'expression suivante :

$$Length(preamble + MHD) + \sum_{k=0}^{t-1} TLM(k), \text{ où } TLM(k) \text{ représente la}$$

longueur de la tuile k .

On utilise ainsi avantageusement les informations contenues dans le marqueur de pointage TLM qui ont été obtenues à partir de la lecture des données d'en-tête principal (étape E103 de la figure 12).

Lors de l'étape E116, on détermine les paquets nécessaires à la reconstruction de la partie du signal spécifiée par le client pour la tuile t considérée.

Une fois ces paquets identifiés, l'étape suivante E117 transmet au deuxième appareil de communication (serveur) une requête visant à obtenir les paquets identifiés qui correspondent à la requête de l'utilisateur.

On notera qu'entre cette étape E117 et l'étape suivante E118, les algorithmes des figures 7 et 9 sont mis en œuvre par le deuxième appareil de communication récepteur de la requête afin de traiter celle-ci, c'est-à-dire de déterminer la position des paquets identifiés dans la requête, de les extraire du signal et de les transmettre au premier appareil de communication.

Au cours de l'étape E118 le premier appareil de communication reçoit et stocke de façon adaptée les paquets ainsi reçus.

L'algorithme de la **figure 14** détaille les opérations effectuées à l'étape E118 de la figure 11 et fait également partie de cet algorithme.

L'algorithme de la figure 14 comporte différentes instructions ou portions de code correspondant à des étapes du procédé de traitement des données reçues selon l'invention.

Cet algorithme débute par une étape E119 d'initialisation de l'indice de paquet i à la valeur 0, afin de traiter tout d'abord le premier paquet de l'ensemble des paquets reçus.

Au cours de l'étape suivante E120 on procède à la détermination de l'ordre d'apparition du paquet de données considéré dans le corps du signal, en fonction de paramètres relatifs à la structure et à l'organisation des données dans le signal.

Ceci s'effectue plus particulièrement en fonction des données d'en-tête du signal et cette étape est identique à l'étape E71 de la figure 9 et ne sera donc pas davantage reprise ici.

5 Dès lors que l'ordre d'apparition du paquet dans le corps du signal, à savoir son numéro de séquence SN_i , a été déterminé, alors l'étape E120 est suivie d'une étape E121.

10 Au cours de cette étape on détermine la position à laquelle doit être inséré le paquet considéré dans le corps du signal dérivé en fonction, d'une part, de la longueur des données d'en-tête et, d'autre part, d'au moins un marqueur de pointage qui a préalablement été reçu dans les données d'en-tête et qui est adapté à fournir la longueur de la partie du corps précédant ce paquet de données.

Pour ce faire, on procède de manière identique à ce qui a été décrit en référence à l'étape E72 de la figure 9.

15 Ainsi, la longueur de la partie du corps du signal précédant le paquet de données considéré ici est déterminée à partir du marqueur de pointage TLM dont une partie fournit la longueur de la ou des tuiles précédant celle où se trouve le paquet considéré.

20 Ainsi, par exemple, comme illustré à la figure 13, si le paquet considéré doit être inséré dans la tuile 1, alors on s'intéresse aux informations fournies par le marqueur de pointage TLM qui donne la longueur de la tuile 0.

25 Par ailleurs, la longueur de la partie du corps du signal précédant le paquet de données considéré est également déterminée à partir de la longueur des données d'en-tête de la tuile où se trouve ce paquet, à savoir, dans l'exemple de la figure 13, les données d'en-tête 158 de la tuile 1.

30 La longueur de la partie du corps du signal précédant le paquet considéré est, en outre, déterminée à partir de valeurs contenues dans le marqueur de pointage PLT qui est adapté à fournir la longueur des paquets de données précédant le paquet considéré dans la tuile où il se trouve, à savoir la tuile 1.

Ainsi, une fois déterminée la position du paquet considéré dans le signal dérivé (stocké dans le fichier de cache), on passe à l'étape suivante

E122 au cours de laquelle on procède à l'insertion, dans le corps du signal dérivé, de ce paquet à la position déterminée à l'étape précédente.

Le nouveau paquet de longueur $PLT_i(SN)$ est ainsi inséré dans le corps de la tuile 1 par écriture dans le fichier de cache des octets constituant ce
5 paquet (figure 13).

Il convient de noter que cette opération d'écriture conduit à écraser une partie de l'espace initialement rempli de données arbitraires et qui étaient destinées à recevoir des données en provenance du deuxième appareil de communication.

10 Au cours de l'étape suivante E123 il est prévu d'effectuer un test afin de déterminer si le paquet qui vient d'être traité est le dernier paquet reçu.

Dans la négative, cette étape est suivie d'une étape E124 au cours de laquelle on incrémente d'une unité l'indice de paquet afin de s'intéresser au prochain paquet de l'ensemble des paquets reçus, et cette étape est alors
15 suivie de l'étape E120 déjà décrite ci-dessus.

Au contraire, lorsque le résultat du test pratiqué à l'étape E123 est positif alors il est mis fin à l'algorithme de la figure 14 par une étape E125.

Dès lors que l'algorithme de la figure 14 a été exécuté, l'étape E118 de la figure 11 est suivie d'une étape E130, au cours de laquelle un test est
20 pratiqué afin de déterminer si la tuile courante t qui vient d'être traitée correspond à la dernière tuile nécessaire pour satisfaire à la requête de l'utilisateur.

Dans la négative, cette étape est suivie d'une étape E131 au cours de laquelle on procède à l'incrément d'une unité de l'indice de tuile, afin de
25 s'intéresser à la prochaine tuile nécessaire déterminée à l'étape E112.

L'étape E131 est ensuite suivie de l'étape E114 déjà décrite ci-dessus.

Dans ce cas, l'étape E114 prévoit de transmettre au deuxième appareil de communication une nouvelle requête visant à obtenir les données d'en-tête de la nouvelle tuile en cours de traitement.

30 Au contraire, lorsque le résultat du test pratiqué à l'étape E130 est positif, alors il est mis fin à l'algorithme de la figure 11 par une étape E132.

Il convient de noter que l'algorithme de la figure 114 prévoit de traiter les paquets reçus par le premier appareil de communication selon le même ordre de progression que celui du train binaire du signal d'image stocké dans le deuxième appareil de communication.

5 Par conséquent, si tous les paquets du signal d'image stocké dans le deuxième appareil de communication sont transférés au premier appareil, alors le signal dérivé est identique au signal d'origine.

Toutefois, il est possible de prévoir une organisation différente des données dans le signal dérivé en fonction, par exemple, des exigences de
10 l'utilisateur.

L'algorithme de la **figure 15** illustre une variante de réalisation qui complète celle décrite en référence à la figure 9. Cette variante concerne le cas où le premier appareil de communication transmet au deuxième appareil de communication une requête visant à obtenir directement une partie du signal
15 présent dans ce deuxième appareil.

Comme déjà indiqué plus haut, cette partie du signal est par exemple précisée dans la requête par la taille et la position de la partie du signal souhaitée, ainsi que le niveau de résolution et de qualité désirés.

L'algorithme de la figure 15 est mis en œuvre par le premier appareil
20 de communication et comporte différentes instructions ou portions de code correspondant à des étapes du procédé de traitement des données reçues par cet appareil.

Cet algorithme débute par une étape E140 de transmission de la requête précitée du premier appareil vers le deuxième appareil.

25 Cette étape est suivie d'une étape E141 au cours de laquelle le premier appareil est en attente d'une réponse R en provenance du deuxième appareil.

Lorsque la réponse R est reçue par le premier appareil, alors on passe à l'étape suivante E142, au cours de laquelle un test est pratiqué afin de
30 déterminer si la réponse contient les données d'en-tête principal du signal stocké dans le deuxième appareil de communication, à savoir le préambule et l'en-tête principal.

Dans l'affirmative, cette étape est suivie d'une étape E143 au cours de laquelle on procède à la formation du signal dérivé et, plus particulièrement, à la formation d'un fichier de cache.

On fait ainsi appel à l'algorithme de la figure 12 pour mettre en œuvre l'étape E143.

On notera que l'étape E143 est identique à l'étape E102 décrite plus en détail en référence à la figure 12 et ne sera donc pas davantage détaillée ici.

L'étape E143 est alors suivie de l'étape E141 précédemment décrite, au cours de laquelle le premier appareil de communication est en attente d'une réponse du deuxième appareil de communication.

De retour à l'étape E142, lorsque le résultat du test pratiqué à cette étape est négatif, alors on passe à l'étape suivante E144 au cours de laquelle un autre test est pratiqué afin de déterminer si la réponse concerne des données d'en-tête de tuile.

Dans l'affirmative, l'étape est suivie d'une étape E145 au cours de laquelle les données d'en-tête de la tuile t considérée sont reçues et insérées à la position déterminée par l'expression suivante :

$$Length(preamble + MHD) + \sum_{k=0}^{t-1} TLM(k), \text{ où } TLM(k) \text{ représente la}$$

longueur de la tuile k .

Cette étape est identique à l'étape E115 de l'algorithme de la figure 11 et ne sera donc pas davantage détaillée ici.

Les données d'en-tête de tuile ayant ainsi été stockées dans le fichier de cache à la position appropriée, l'étape E145 est suivie d'une étape E146 au cours de laquelle on procède à la lecture des données d'en-tête de tuile, afin d'en extraire les informations de longueurs des paquets contenus dans cette tuile à partir du marqueur de pointage PLT présent dans ces données d'en-tête.

L'étape E146 est ensuite suivie de l'étape E141 d'attente d'une réponse en provenance du deuxième appareil de communication et qui a déjà été décrite ci-dessus.

De retour à l'étape E144, lorsque le résultat du test pratiqué à cette étape est négatif, alors celle-ci est suivie d'une étape E147 au cours de laquelle

un autre test est pratiqué afin de déterminer si la réponse provenant du deuxième appareil de communication comporte un ou plusieurs paquets de données.

5 Dans l'affirmative, cette étape est suivie de l'étape E148 au cours de laquelle il est prévu d'identifier les paquets de données reçus et leurs paramètres t_i , c_i , r_i , p_i , l_i , avec $i = 0$ à $nb_{\text{paquets}}-1$.

Cette étape est suivie d'une étape E149 qui est identique à l'étape E118 de l'algorithme de la figure 11 et fait, comme elle, appel à l'algorithme de la figure 14 pour son exécution.

10 Au cours de cette étape il est ainsi prévu, d'une part, de déterminer la position à laquelle les différents paquets reçus dans la réponse doivent être insérés dans le corps du signal dérivé et, d'autre part, de procéder à leur stockage dans ce dernier.

L'étape E149 est ensuite suivie de l'étape E141 déjà décrite ci-dessus.

15 De retour à l'étape E147, lorsque le résultat du test pratiqué à cette étape est négatif, alors celle-ci est suivie d'une étape E150 au cours de laquelle un autre test est pratiqué afin de déterminer si la réponse reçue correspond à un signal de fin de réponse.

20 Dans la négative, alors l'étape E150 est suivie d'une étape E151 au cours de laquelle un signal d'erreur est transmis au deuxième appareil de communication, pour l'informer de ce que la réponse que le premier appareil vient de recevoir est de type inconnu.

Cette étape est ensuite suivie de l'étape E141 déjà décrite ci-dessus.

25 Au contraire, lorsque le résultat du test pratiqué à l'étape E150 est positif, alors il est mis fin à l'algorithme de la figure 15 par une étape E152.

On notera qu'il est avantageux de stocker en mémoire le signal dérivé dans le fichier de cache car, d'une session de communication à l'autre, il est possible de continuer à recevoir des paquets de données concernant ce même signal.

30 La **figure 16** illustre un exemple de signal dérivé formé à partir des données d'en-tête 150 du signal d'origine, dont la structure est représentée à la

figure 6, et de certains paquets de données contenus dans le corps de ce signal.

5 Ainsi, en mettant en œuvre les mécanismes précédemment explicités de détermination de la position des paquets reçus en provenance du deuxième appareil de communication et d'insertion de ces paquets à la position déterminée dans le signal, on obtient, par exemple, la structure représentée en partie haute de la figure 16:

10 Comme représenté dans cette structure, le fichier de cache contenant le signal dérivé ainsi formé comporte dans l'espace initialement rempli de données arbitraires et qui était destiné à contenir les paquets reçus du deuxième appareil, tous les paquets de données reçus de ce deuxième appareil, ainsi que des sous-espaces toujours remplis de données arbitraires et pour lesquels aucun paquet n'a été reçu.

15 Ces sous-espaces sont référencés 190, 192, 194, 196, 198, 200, 202 et 204 sur la figure 16.

Ainsi, le signal dérivé qui vient d'être formé peut-être conservé tel quel en mémoire cache du premier appareil de communication, afin de pouvoir servir, lors de sessions de communication futures, à recevoir d'autres paquets de données concernant ce même signal.

20 Toutefois, il peut également être avantageux de transformer ce signal dérivé en un signal appelé signal valide et qui est valide en ce sens qu'il est conforme à la syntaxe de description prévue dans le standard JPEG2000.

Ceci n'est bien entendu valable que dans le cas où l'invention s'applique aux signaux d'image qui sont conformes aux standard JPEG2000.

25 Dans le cas d'un signal qui n'est pas nécessairement un signal d'image, il peut également être décidé de transformer le signal dérivé formé selon l'invention afin de le rendre conforme au standard auquel se conforme le signal d'origine présent dans le deuxième appareil.

30 La partie inférieure de la figure 16 représente la structure du signal valide issu de la transformation précitée.

Le signal ainsi transformé contient alors autant de tuiles, de composantes, de niveaux de résolution et de couches de qualité que le signal d'origine présent dans le deuxième appareil de communication.

On notera que toutes les données d'en-tête ainsi que les différents
5 paquets de données reçus et insérés dans le signal dérivé représenté en partie haute de la figure 16 ont été extraits de ce signal, les données d'en-tête ayant servi à former les données d'en-tête du signal valide.

En ce qui concerne les paquets de données extraits, ceux-ci ont été concaténés dans le corps du signal valide.

10 Là où des paquets de données n'ont pas été reçus, on a concaténé à la suite des autres paquets, dans le signal valide, des paquets vides suivant le même ordre d'apparition que celui adopté dans le signal dérivé.

On notera que les paquets vides ont chacun une longueur égale à au plus un octet.

15 Les données d'en-tête principal et les données d'en-tête de tuiles sont ainsi légèrement modifiées afin de tenir compte de la diminution de longueur de chaque tuile due à l'insertion de paquets vides.

La création de tels paquets vides dans le signal valide constitue un moyen simple de construire un signal valide au sens du standard JPEG2000.

20 Avantageusement, un tel signal valide permet de pouvoir exploiter ultérieurement l'ensemble des données obtenues lors de la session de communication entre le premier et le deuxième appareil de communication, à l'aide d'un décodeur conforme au standard JPEG2000 et qui est indépendant de l'architecture client-serveur utilisée.

25 On va maintenant décrire plus en détail l'algorithme de transformation du signal dérivé en un signal valide en référence à la figure 17.

Cet algorithme comporte différentes instructions ou portions de code correspondant à des étapes du procédé de traitement des données reçues par le premier appareil de communication selon l'invention.

30 On notera que le programme d'ordinateur "Progr" stocké dans l'appareil de communication représenté à la figure 2 et qui correspond ici au

premier appareil de communication selon l'invention est également basé sur cet algorithme.

5 L'algorithme de la figure 17 débute par une étape E160 au cours de laquelle les données d'en-tête principal 152 et 154 (figure 16) sont extraites du signal dérivé et recopiées dans un nouveau fichier dit valide.

On notera qu'ainsi on forme les données d'en-tête du signal valide à partir des données d'en-tête extraites du signal dérivé.

Il convient également de noter que, lors de cette étape, les données d'en-tête principal sont quelque peu modifiées afin de simplifier l'algorithme.

10 En effet, si un marqueur de pointage PLM fournissant les longueurs des paquets de chacune des tuiles dans le signal est présent dans les données d'en-tête principal, alors il n'est pas reproduit dans le signal valide.

Ceci permet d'éviter d'avoir à modifier ce marqueur ultérieurement, lors de la concaténation de paquets vides dans le signal valide.

15 On notera, par ailleurs, que le marqueur de pointage TLM présent dans les données d'en-tête principal peut être maintenu ou non dans les données d'en-tête principal du signal valide.

En effet, le maintien de ce marqueur va nécessiter de le modifier vers la fin de l'algorithme, mais ce maintien est justifié dans la mesure où la
20 modification envisagée est peu complexe à réaliser.

L'étape suivante E161 prévoit de se positionner à la première tuile 0 du signal, celle-ci fournissant alors la valeur de l'indice de tuile courante t.

On va ainsi parcourir toutes les tuiles du signal une à une.

25 Cette étape E161 est suivie d'une étape E162 au cours de laquelle un test est pratiqué afin de déterminer si les données d'en-tête de la tuile t concernée ont été reçues par le premier appareil de communication.

Dans l'affirmative, l'étape est suivie d'une étape E163 au cours de laquelle ces données d'en-tête de tuile sont insérées dans le signal valide que l'on vient de commencer à former à l'étape E160, par copie de ces données
30 d'en-tête dans le fichier valide contenant ce signal.

Toutefois, le marqueur de pointage PLT contenu dans ces données d'en-tête de tuile n'est pas maintenu pour des raisons identiques à celles mentionnées ci-dessus pour le marqueur PLM.

5 L'étape E163 est ensuite suivie d'une étape E164 au cours de laquelle on se positionne sur le premier paquet de données de la tuile t considérée, et l'indice de paquets sn est alors initialisé à la valeur 0 correspondant à ce premier paquet.

De retour à l'étape E162, lorsque le résultat du test pratiqué à cette étape est négatif, alors on passe à l'étape suivante E165, au cours de laquelle
10 on insère, dans le signal valide, des marqueurs SOT et SOD, ce qui revient à insérer des données d'en-tête de tuiles vides.

Du point de vue de la syntaxe de description JPEG2000, ceci signifie que le corps de la tuile considérée contiendra autant de paquets que l'indiquent les données d'en-tête principal du signal valide.

15 Dans la mesure où aucun paquet n'a été reçu dans la tuile considérée, tous les paquets constituant le corps de cette tuile seront donc vides.

L'étape E165 est ensuite suivie de l'étape E164 précédemment décrite.

L'étape E164 est suivie à son tour d'une étape E166, au cours de
20 laquelle un test est pratiqué, afin de déterminer si le paquet d'indice sn a été reçu et inséré dans le signal dérivé représenté en partie haute de la figure 16.

Dans l'affirmative, cette étape est suivie d'une étape E167 au cours de laquelle les PLT_{t_i} (SN_i) octets qui commencent à la position "Position(t_i , SN_i)" dans le signal dérivé et correspondent aux octets du paquet sn reçu sont
25 insérés dans le corps de ce signal et, plus particulièrement, concaténés avec les autres données constitutives des autres paquets déjà présents.

Cette étape est ensuite suivie d'une étape E168 au cours de laquelle un test est pratiqué, afin de déterminer si l'indice de paquet sn correspond à l'indice du dernier paquet de la tuile courante.

30 Dans la négative, on passe à l'étape E169 au cours de laquelle l'indice de paquet est incrémenté d'une unité et cette étape est alors suivie de l'étape E166 précédemment décrite.

Lorsque le résultat du test pratiqué à l'étape E166 est négatif, alors cela signifie que le paquet d'indice sn n'a pas été reçu et l'étape suivante E170 prévoit d'insérer un paquet vide dans le corps du signal valide et, plus particulièrement, de concaténer un paquet vide à la suite des paquets déjà en place.

On notera qu'un paquet vide est signalé par le premier bit de son en-tête qui est égal à 0.

Il convient de noter que si un bit à 0 est présent dans le signal pour indiquer un paquet vide, alors il est éventuellement complété afin que l'en-tête de paquet se termine à la frontière d'un octet, comme prévu par le standard JPEG2000.

L'étape E170 est alors suivie de l'étape E168 déjà décrite ci-dessus.

Lorsque le résultat du test pratiqué à l'étape E168 est positif, ceci signifie que tous les paquets de la tuile courante ont été concaténés dans le signal valide.

On passe alors à l'étape suivante E171 au cours de laquelle la longueur totale de la tuile courante est calculée et le champ Psot du marqueur SOT des données d'en-tête de la tuile est modifié afin d'indiquer la longueur totale de cette tuile.

On notera que le codage de ce champ est connu de l'homme de l'art et défini dans la partie I du standard JPEG2000 dont les références complètes ont été fournies plus haut.

L'étape E171 est ensuite suivie d'une étape E172 au cours de laquelle un test est pratiqué afin de déterminer si l'indice de tuile t correspond à l'indice de la dernière tuile du signal d'image considéré.

Dans la négative, cette étape est suivie d'une étape E173 qui prévoit d'incrémenter d'une unité l'indice de la tuile et est alors suivie de l'étape E162 déjà décrite ci-dessus.

Dans l'affirmative, l'étape E172 est suivie d'une étape E174 au cours de laquelle un test est pratiqué afin de déterminer si le marqueur de pointage TLM est présent dans les données d'en-tête principal.

Dans l'affirmative, l'étape suivante E175 prévoit de recoder les différents champs Ptlm (notés aussi TLM (k), où k est l'indice de tuile) de ce marqueur et qui fournissent les longueurs respectives des différentes tuiles du signal d'image.

5 On notera au passage que, pour ce faire, il est nécessaire d'avoir mémorisé au préalable les longueurs de chaque tuile dans le signal valide en cours de formation.

Cette opération de recodage est là encore également connue de l'homme de l'art.

10 Il convient de noter que le recodage des champs Ptlm du marqueur de pointage TLM n'implique aucune modification de la longueur totale de ce marqueur.

Ainsi, l'opération de recodage ne nécessite pas d'insérer ou de supprimer des octets dans le signal valide en cours de formation.

15 L'étape E175 est ensuite suivie d'une étape E176 qui prévoit d'écrire le marqueur 172 noté EOC et indiquant la fin du signal valide.

De retour à l'étape E174, lorsque le résultat du test pratiqué est négatif, alors on passe directement à l'étape E176 qui vient d'être décrite.

Il est ensuite mis fin à l'algorithme par une étape E177.

20 L'algorithme de la **figure 18** propose une variante de réalisation de la transformation du signal valide illustrée par l'algorithme de la figure 17.

A la différence de ce dernier qui forme un nouveau train binaire dans un nouveau fichier afin que celui-ci soit valide au sens du standard JPEG2000, le mécanisme proposé à la figure 18 envisage de transformer le signal dérivé
25 qui est stocké dans le fichier de cache, afin de le rendre conforme au standard JPEG2000 à l'intérieur de ce même fichier.

Cet algorithme comporte différentes instructions ou portions de code correspondant à des étapes du procédé selon l'invention.

30 Cet algorithme débute par une étape E180 au cours de laquelle on initialise l'indice de tuile t à la première tuile 0.

Au cours de cette étape, on se positionne également dans le signal dérivé à l'endroit où vont être effectuées les transformations du signal.

Cette position initiale correspond au début du marqueur de pointage TLM dans le cas où l'on souhaite que ce marqueur soit maintenu dans la version valide du signal dérivé.

Si par contre il est décidé de ne pas le maintenir, alors la position
5 initiale est située juste après la fin de ce marqueur.

L'étape suivante E182 prévoit d'effectuer un test afin de déterminer si les données d'en-tête de la tuile t considérée ont été reçues et sont présentes dans le signal dérivé.

Dans l'affirmative, on passe à l'étape E183 au cours de laquelle il est
10 prévu d'écrire les données d'en-tête de tuile, à l'exception du marqueur PLT, dans le signal, à la position courante correspondant, initialement, à la position indiquée à l'étape E180.

On notera que pour la première tuile ainsi traitée, les données d'en-tête de tuile sont déjà présentes à la bonne position.

15 Par contre, lorsque l'étape E183 sera exécutée pour une autre tuile et que des paquets de données n'auront pas été reçues dans la tuile précédente, alors, effectivement, il sera nécessaire de décaler de façon adaptée les données et donc de réécrire ces données d'en-tête de tuile à la bonne position.

L'étape E183 est ensuite suivie d'une étape E184 au cours de laquelle
20 il est prévu de se positionner sur le premier paquet de la tuile t considérée, l'indice du premier paquet sn étant initialisé à la valeur 0.

De retour à l'étape E182, lorsque le résultat du test pratiqué à cette étape est négatif, alors on passe à l'étape E185 qui est identique à l'étape E165 de la figure 17 et, au cours de laquelle, on écrit dans le signal des données
25 d'en-tête de tuiles vides, c'est-à-dire un marqueur SOT et un marqueur SOD.

L'étape E185 est ensuite suivie de l'étape E184 précédemment décrite.

Cette dernière est suivie d'une étape E186 au cours de laquelle un test est pratiqué afin de déterminer si le paquet d'indice sn a été reçu.

30 Dans l'affirmative, ce paquet est écrit octet par octet à la position courante considérée pos (étape E187).

Cette étape est identique à l'étape E167 de l'algorithme de la figure 17.

Il convient de noter qu'à chaque fois qu'on écrit un octet dans le signal, la position courante pos est incrémentée d'une unité.

5 L'étape E187 est ensuite suivie d'une étape E188 au cours de laquelle un test est pratiqué afin de déterminer si l'indice sn du paquet correspond à l'indice du dernier paquet de la tuile courante t considérée.

Dans la négative, cette étape est suivie de l'étape E189 qui incrémente d'une unité cet indice, afin de passer au paquet suivant et est suivie
10 de l'étape E186 précédemment décrite.

Lorsque le résultat du test pratiqué à l'étape E186 est négatif, ceci signifie que le paquet d'indice sn n'a pas été reçu et on écrit alors un paquet vide dans le signal lors de l'étape suivante E190.

Il convient de noter que l'on écrit plus particulièrement l'octet 0 x 80 en
15 notation hexa décimale.

L'étape E190 est ensuite suivie de l'étape E188 précédemment décrite.

Comme déjà décrit en référence à la figure 17, les étapes suivantes E191, E192, E193, E194, E195 et E196 consistent à mettre à jour le marqueur
20 de pointage TLM, s'il a été maintenu dans la version valide du signal dérivé, et à écrire le marqueur EOC de fin de signal.

Les étapes qui viennent d'être citées correspondent respectivement aux étapes E171, E172, E173, E174, E175 et E176 de l'algorithme de la figure 17 et ne seront donc pas davantage décrites ici.

25 Par contre, l'algorithme de la figure 18 comporte, consécutivement à l'étape E196, une étape E197 au cours de laquelle il est prévu de supprimer les octets restants du signal qui vient d'être transformé comme indiqué lors des étapes précédentes.

En effet, le signal dérivé qui a initialement été formé est de longueur
30 supérieure à celle du signal valide ainsi formé.

Il est ensuite mis fin à l'algorithme de la figure 18 par l'étape E198.

Il convient de noter que l'écriture d'un octet dans l'algorithme de la figure 18 consiste à écraser l'octet déjà présent à la même position dans le signal en cours de formation.

- 5 Par ailleurs, dans la plupart des cas, l'algorithme proposé évite d'avoir à mémoriser des données du signal dérivé à transformer dans le but de ne pas écraser de données utiles qui ont été reçues.

REVENDICATIONS

1. Procédé de traitement d'une requête provenant d'un premier
appareil de communication connecté à travers un réseau de communication à
5 un deuxième appareil de communication distant, le procédé comportant une
étape de réception de la requête d'obtention de données numériques d'un
signal numérique compressé qui comporte des données d'en-tête et un corps
de signal comprenant notamment des paquets de données, caractérisé en ce
que le traitement de la requête comporte une étape de détermination de la
10 position, dans le corps du signal, d'au moins un paquet de données
correspondant à la requête en fonction, d'une part, de la longueur des données
d'en-tête et, d'autre part, d'au moins un marqueur de pointage présent dans le
signal et adapté à fournir la longueur de la partie du corps précédant le paquet
de données considéré.

15 2. Procédé selon la revendication 1, caractérisé en ce que la
détermination de la longueur de la partie du corps du signal précédant le paquet
de données considéré comporte une étape préalable de détermination de
l'ordre d'apparition dudit paquet de données dans le corps du signal, en fonction
de paramètres relatifs à la structure et à l'organisation des données dans le
20 signal.-

3. Procédé selon la revendication 1 ou 2, caractérisé en ce que le
signal numérique compressé est partitionné en un nombre n de zones t_i
compressées de façon indépendante, $i = 1$ à n et $n \geq 1$, le corps du signal
comprenant pour chaque zone des données d'en-tête de zone et un corps de
25 zone contenant des paquets de données de la zone considérée.

4. Procédé selon la revendication 3, caractérisé en ce que la
longueur de la partie du corps du signal précédant le paquet de données
considéré est déterminée à partir de :

- au moins un marqueur de pointage (PLT) adapté à fournir
30 notamment la longueur du ou des paquets de données précédant le paquet de
données considéré dans la zone où se trouve ce paquet,

- la longueur des données d'en-tête de la zone où se trouve le paquet considéré et, lorsqu'une ou plusieurs zones précèdent la zone où se trouve le paquet considéré,

5 - au moins un marqueur de pointage (TLM) adapté à fournir notamment la longueur de la ou des zones précédentes.

5. Procédé selon la revendication 4, caractérisé en ce que le marqueur de pointage (TLM) adapté à fournir la longueur de chaque zone t_i est présent dans les données d'en-tête.

10 6. Procédé selon la revendication 4, caractérisé en ce que le marqueur de pointage (PLT) adapté à fournir la longueur des paquets de données dans une zone t_i est présent dans les données d'en-tête de la zone concernée.

15 7. Procédé selon l'une des revendications 1 à 6, caractérisé en ce qu'il comporte les étapes d'extraction et de transmission au premier appareil de communication dudit au moins un paquet de données dont la position a été déterminée.

20 8. Procédé selon l'une des revendications 1 à 7, caractérisé en ce que la requête d'obtention de données numériques spécifie au moins un paquet de données du signal.

25 9. Procédé selon l'une des revendications 1 à 7, caractérisé en ce que la requête d'obtention de données numériques spécifie une partie du signal.

30 10. Procédé selon la revendication 9, caractérisé en ce que, postérieurement à la réception de la requête, le procédé comporte une étape d'identification du ou des paquets de données nécessaires à la reconstruction de la partie du signal spécifiée.

11. Procédé selon l'une des revendications 1 à 10, caractérisé en ce qu'il comporte une étape préalable de formation dudit au moins un marqueur de pointage dans le signal, lorsqu'un tel marqueur n'est pas présent dans le signal.

12. Procédé de traitement de données numériques compressées reçues par un premier appareil de communication connecté à travers un réseau de communication à un deuxième appareil de communication distant, le

procédé comportant une étape de réception d'au moins un paquet de données provenant d'un signal numérique compressé présent dans le deuxième appareil et comportant un corps qui comprend notamment des paquets de données, caractérisé en ce que le procédé comporte les étapes suivantes :

- 5 - détermination d'une position à laquelle doit être inséré ledit au moins un paquet de données dans le corps d'un signal numérique compressé dérivé du signal numérique compressé présent dans le deuxième appareil et qui est apte à contenir tout ou partie du corps de ce signal numérique compressé, le signal dérivé comportant également des données d'en-tête, la détermination de
10 la position étant effectuée en fonction, d'une part, de la longueur des données d'en-tête et, d'autre part, d'au moins un marqueur de pointage préalablement reçu et inséré dans le signal par le premier appareil et qui est adapté à fournir la longueur de la partie du corps précédant ledit au moins un paquet de données,
 - insertion dans le corps du signal dérivé dudit au moins un paquet de
15 données à la position ainsi déterminée.

13. Procédé selon la revendication 12, caractérisé en ce qu'il comporte les étapes préalables suivantes :

- réception des données d'en-tête provenant du signal numérique compressé d'origine présent dans le deuxième appareil, les données d'en-tête
20 reçues comprenant au moins un marqueur de pointage (TLM) adapté à fournir la longueur du corps du signal d'origine,
 - à partir des données d'en-tête reçues, formation du signal numérique compressé dérivé qui comprend ainsi, comme données d'en-tête, les données d'en-tête reçues et un corps de signal de longueur égale à celle du corps du
25 signal d'origine, le corps du signal dérivé représentant un espace initialement rempli de données arbitraires et qui est destiné à contenir le ou les paquets de données reçus du deuxième appareil.

14. Procédé selon la revendication 12 ou 13, caractérisé en ce que le signal numérique compressé est partitionné en un nombre de zones t_i
30 compressées de façon indépendante, $i = 1$ à n et $n \geq 1$, le corps du signal comprenant pour chaque zone des données d'en-tête de zone et un corps de zone contenant des paquets de données de la zone considérée.

15. Procédé selon la revendication 14, caractérisé en ce que la longueur de la partie du corps du signal précédant le paquet de données considéré est déterminée à partir de :

- au moins un marqueur de pointage (PLT) adapté à fournir
5 notamment la longueur du ou des paquets de données précédant le paquet de données considéré dans la zone où se trouve ce paquet,
- la longueur des données d'en-tête de la zone où se trouve le paquet considéré et, lorsqu'une ou plusieurs zones précèdent la zone où se trouve le paquet considéré,
- 10 - au moins un marqueur de pointage (TLM) adapté à fournir notamment la longueur de la ou des zones précédentes.

16. Procédé selon la revendication 15, caractérisé en ce que le marqueur de pointage adapté à fournir la longueur de chaque zone t_i est présent dans les données d'en-tête.

- 15 17. Procédé selon la revendication 15, caractérisé en ce que le marqueur de pointage adapté à fournir la longueur des paquets de données dans une zone t_i est présent dans les données d'en-tête de la zone concernée.

18. Procédé selon l'une des revendications 14 à 17, caractérisé en ce qu'il comporte les étapes suivantes :

- 20 - réception de données d'en-tête de zone,
- détermination d'une position à laquelle doivent être insérées les données d'en-tête de zone reçues dans le corps du signal dérivé, la détermination de la position étant effectuée en fonction de la longueur des données d'en-tête du signal dérivé et, lorsqu'une ou plusieurs zones précèdent
25 les données d'en-tête de zone concernées, également en fonction d'un ou plusieurs marqueurs de pointage (TLM) préalablement reçus et fournissant respectivement la longueur de la ou des zones précédentes,
- insertion des données d'en-tête de zone reçues à la position ainsi déterminée.

- 30 19. Procédé selon l'une des revendications 12 à 18, caractérisé en ce que la détermination de la longueur de la partie du corps du signal dérivé précédant le paquet de données considéré comporte une étape préalable de

détermination de l'ordre d'apparition dudit paquet de données dans le corps du signal en fonction de paramètres relatifs à la structure et à l'organisation des données dans le signal.

20. Procédé selon la revendication 13, caractérisé en ce qu'il
5 comporte une phase de transformation du signal dérivé en un signal valide qui comporte les étapes suivantes :

- extraction du signal dérivé des données d'en-tête et des paquets de données reçus,
- formation des données d'en-tête du signal valide à partir des
10 données d'en-tête extraites du signal dérivé,
- concaténation des paquets de données extraits du signal dérivé dans le corps du signal valide, et
- lorsqu'un ou plusieurs paquets de données présents dans le corps du signal d'origine ne sont pas reçus par le premier appareil, concaténation
15 respectivement d'un ou de plusieurs paquets vides dans le corps du signal valide suivant le même ordre d'apparition que celui adopté dans le signal dérivé.

21. Procédé selon la revendication 13, caractérisé en ce qu'il
comporte les étapes suivantes :

- 20 - parcours des données contenues dans le corps du signal dérivé,
- lorsque les données parcourues ne correspondent pas à un paquet de données reçu du deuxième appareil, transformation de l'espace rempli par les données concernées en un paquet vide et,
- décalage de façon adaptée des données constituant la suite du
25 corps du signal dérivé.

22. Procédé selon l'une des revendications 12 à 21, caractérisé en ce que les données reçues par le premier appareil constituent la réponse à une requête préalablement transmise du premier appareil au deuxième appareil.

23. Dispositif de traitement d'une requête provenant d'un premier
30 appareil de communication connecté à travers un réseau de communication à un deuxième appareil de communication distant, le dispositif comportant des moyens de réception de la requête d'obtention de données numériques d'un

signal numérique compressé qui comporte des données d'en-tête et un corps de signal comprenant notamment des paquets de données, caractérisé en ce que le dispositif comporte, pour le traitement de la requête, des moyens de détermination de la position, dans le corps du signal, d'au moins un paquet de données correspondant à la requête en fonction, d'une part, de la longueur des données d'en-tête et, d'autre part, d'au moins un marqueur de pointage présent dans le signal et adapté à fournir la longueur de la partie du corps précédant le paquet de données considéré.

24. Dispositif selon la revendication 23, caractérisé en ce que les moyens de détermination de la longueur de la partie du corps du signal précédant le paquet de données considéré comportent notamment des moyens de détermination de l'ordre d'apparition dudit paquet de données dans le corps du signal, en fonction de paramètres relatifs à la structure et à l'organisation des données dans le signal.

25. Dispositif selon la revendication 23 ou 24, caractérisé en ce que le signal numérique compressé est partitionné en un nombre n de zones t_i compressées de façon indépendante, $i = 1$ à n et $n \geq 1$, le corps du signal comprenant pour chaque zone des données d'en-tête de zone et un corps de zone contenant des paquets de données de la zone considérée.

26. Dispositif selon la revendication 25, caractérisé en ce que la longueur de la partie du corps du signal précédant le paquet de données considéré est déterminée à partir de :

- au moins un marqueur de pointage (PLT) adapté à fournir notamment la longueur du ou des paquets de données précédant le paquet de données considéré dans la zone où se trouve ce paquet,

- la longueur des données d'en-tête de la zone où se trouve le paquet considéré et, lorsqu'une ou plusieurs zones précèdent la zone où se trouve le paquet considéré,

- au moins un marqueur de pointage (TLM) adapté à fournir notamment la longueur de la ou des zones précédentes.

27. Dispositif selon l'une des revendications 23 à 26, caractérisé en ce qu'il comporte des moyens d'extraction et de transmission au premier

appareil de communication dudit au moins un paquet de données dont la position a été déterminée.

28. Dispositif selon l'une des revendications 23 à 27, caractérisé en ce qu'il comporte des moyens de formation dudit au moins un marqueur de pointage dans le signal, lorsqu'un tel marqueur n'est pas présent dans le signal.

29. Dispositif de traitement de données numériques compressées reçues par un premier appareil de communication connecté à travers un réseau de communication à un deuxième appareil de communication distant, le dispositif comportant des moyens de réception d'au moins un paquet de données provenant d'un signal numérique compressé présent dans le deuxième appareil et comportant un corps qui comprend notamment des paquets de données, caractérisé en ce que le dispositif comporte :

- des moyens de détermination d'une position à laquelle doit être inséré ledit au moins un paquet de données dans le corps d'un signal numérique compressé dérivé du signal numérique compressé présent dans le deuxième appareil et qui est apte à contenir tout ou partie du corps de ce signal numérique compressé, le signal dérivé comportant également des données d'en-tête, la détermination de la position étant effectuée en fonction, d'une part, de la longueur des données d'en-tête et, d'autre part, d'au moins un marqueur de pointage préalablement reçu et inséré dans le signal par le premier appareil et qui est adapté à fournir la longueur de la partie du corps précédant ledit au moins un paquet de données,

- des moyens d'insertion dans le corps du signal dérivé dudit au moins un paquet de données à la position ainsi déterminée.

30. Dispositif selon la revendication 29, caractérisé en ce qu'il comporte :

- des moyens de réception des données d'en-tête provenant du signal numérique compressé d'origine présent dans le deuxième appareil, les données d'en-tête reçues comprenant au moins un marqueur de pointage (TLM) adapté à fournir la longueur du corps du signal d'origine,

- des moyens de formation du signal numérique compressé dérivé à partir des données d'en-tête reçues et qui comprend ainsi, comme données

d'en-tête, les données d'en-tête reçues et un corps de signal de longueur égale à celle du corps du signal d'origine, le corps du signal dérivé représentant un espace initialement rempli de données arbitraires et qui est destiné à contenir le ou les paquets de données reçus du deuxième appareil.

5 31. Dispositif selon la revendication 29 ou 30, caractérisé en ce que le signal numérique compressé est partitionné en un nombre n de zones t_i compressées de façon indépendante, $i = 1$ à n et $n \geq 1$, le corps du signal comprenant pour chaque zone des données d'en-tête de zone et un corps de zone contenant des paquets de données de la zone considérée.

10 32. Dispositif selon la revendication 31, caractérisé en ce que la longueur de la partie du corps du signal précédant le paquet de données considéré est déterminée à partir de :

15 - au moins un marqueur de pointage (PLT) adapté à fournir notamment la longueur du ou des paquets de données précédant le paquet de données considéré dans la zone où se trouve ce paquet,

 - la longueur des données d'en-tête de la zone où se trouve le paquet considéré et, lorsqu'une ou plusieurs zones précèdent la zone où se trouve le paquet considéré,

20 - au moins un marqueur de pointage (TLM) adapté à fournir notamment la longueur de la ou des zones précédentes.

 33. Dispositif selon la revendication 31 ou 32, caractérisé en ce qu'il comporte :

25 - des moyens de réception de données d'en-tête de zone,
 - des moyens de détermination d'une position à laquelle doivent être insérées les données d'en-tête de zone reçues dans le corps du signal dérivé, la détermination de la position étant effectuée en fonction de la longueur des données d'en-tête du signal dérivé et, lorsqu'une ou plusieurs zones précèdent les données d'en-tête de zone concernées, également en fonction d'un ou plusieurs marqueurs de pointage (TLM) préalablement reçus et fournissant
30 respectivement la longueur de la ou des zones précédentes,

 - des moyens d'insertion des données d'en-tête de zone reçues à la position ainsi déterminée.

34. Dispositif selon l'une des revendications 29 à 33, caractérisé en ce que les moyens de détermination de la longueur de la partie du corps du signal dérivé précédant le paquet de données considéré comportent notamment des moyens de détermination de l'ordre d'apparition dudit paquet de données dans le corps du signal, en fonction de paramètres relatifs à la structure et à l'organisation des données dans le signal.

35. Procédé selon la revendication 30, caractérisé en ce qu'il comporte des moyens de transformation du signal dérivé en un signal valide qui comportent plus particulièrement :

- des moyens d'extraction du signal dérivé des données d'en-tête et paquets de données reçus,
- des moyens de formation des données d'en-tête du signal valide à partir des données d'en-tête extraites du signal dérivé,
- des moyens de concaténation des paquets de données extraits du signal dérivé dans le corps du signal valide et, lorsqu'un ou plusieurs paquets de données présents dans le corps du signal d'origine ne sont pas reçus par le premier appareil, de concaténation respectivement d'un ou de plusieurs paquets vides dans le corps du signal valide suivant le même ordre d'apparition que celui adopté dans le signal dérivé.

36. Dispositif selon la revendication 30, caractérisé en ce qu'il comporte :

- des moyens de parcours des données contenues dans le corps du signal dérivé,
- lorsque les données parcourues ne correspondent pas à un paquet de données reçu du deuxième appareil, des moyens de transformation de l'espace rempli par les données concernées en un paquet vide et,
- des moyens de décalage de façon adaptée des données constituant la suite du corps du signal dérivé.

37. Appareil de communication, caractérisé en ce qu'il comporte un dispositif de traitement d'une requête selon l'une des revendications 23 à 28.

38. Appareil de communication, caractérisé en ce qu'il comporte un dispositif de traitement de données selon l'une des revendications 29 à 36.

34. Dispositif selon l'une des revendications 29 à 33, caractérisé en ce que les moyens de détermination de la longueur de la partie du corps du signal dérivé précédant le paquet de données considéré comportent notamment des moyens de détermination de l'ordre d'apparition dudit paquet de données dans le corps du signal, en fonction de paramètres relatifs à la structure et à l'organisation des données dans le signal.

35. Dispositif selon la revendication 30, caractérisé en ce qu'il comporte des moyens de transformation du signal dérivé en un signal valide qui comportent plus particulièrement :

- des moyens d'extraction du signal dérivé des données d'en-tête et paquets de données reçus,
- des moyens de formation des données d'en-tête du signal valide à partir des données d'en-tête extraites du signal dérivé,
- des moyens de concaténation des paquets de données extraits du signal dérivé dans le corps du signal valide et, lorsqu'un ou plusieurs paquets de données présents dans le corps du signal d'origine ne sont pas reçus par le premier appareil, de concaténation respectivement d'un ou de plusieurs paquets vides dans le corps du signal valide suivant le même ordre d'apparition que celui adopté dans le signal dérivé.

36. Dispositif selon la revendication 30, caractérisé en ce qu'il comporte :

- des moyens de parcours des données contenues dans le corps du signal dérivé,
- lorsque les données parcourues ne correspondent pas à un paquet de données reçu du deuxième appareil, des moyens de transformation de l'espace rempli par les données concernées en un paquet vide et,
- des moyens de décalage de façon adaptée des données constituant la suite du corps du signal dérivé.

37. Appareil de communication, caractérisé en ce qu'il comporte un dispositif de traitement d'une requête selon l'une des revendications 23 à 28.

38. Appareil de communication, caractérisé en ce qu'il comporte un dispositif de traitement de données selon l'une des revendications 29 à 36.

39. Moyen de stockage d'informations lisible par un ordinateur ou un microprocesseur comportant des instructions de code d'un programme d'ordinateur pour l'exécution des étapes du procédé de traitement d'une requête selon l'une des revendications 1 à 11.

5 40. Moyen de stockage d'informations lisible par un ordinateur ou un microprocesseur comportant des instructions de code d'un programme d'ordinateur pour l'exécution des étapes du procédé de traitement de données selon l'une des revendications 12 à 22.

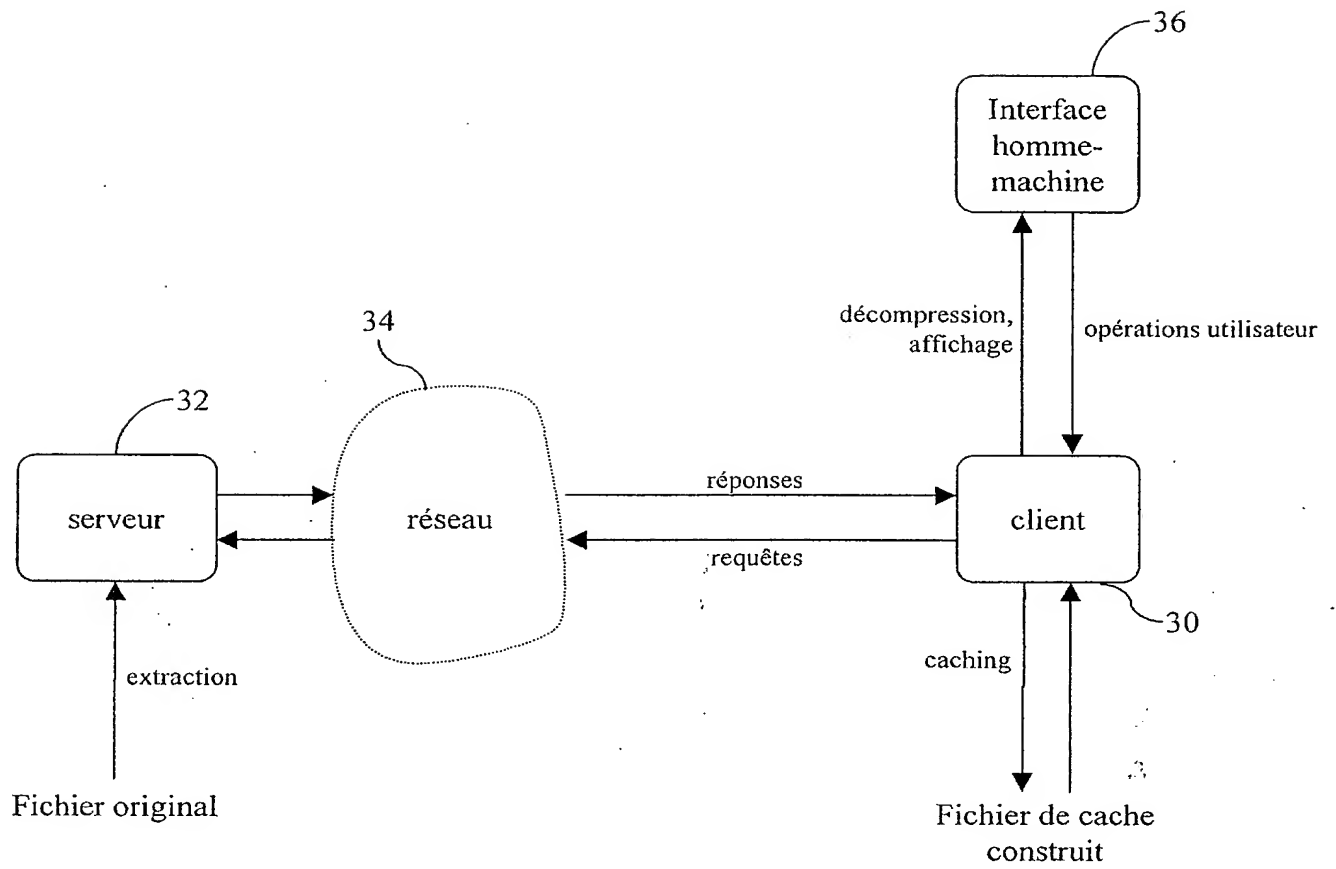
10 41. Moyen de stockage d'informations amovible, partiellement ou totalement lisible par un ordinateur ou un microprocesseur comportant des instructions de code d'un programme d'ordinateur pour l'exécution des étapes du procédé de traitement d'une requête selon l'une des revendications 1 à 11.

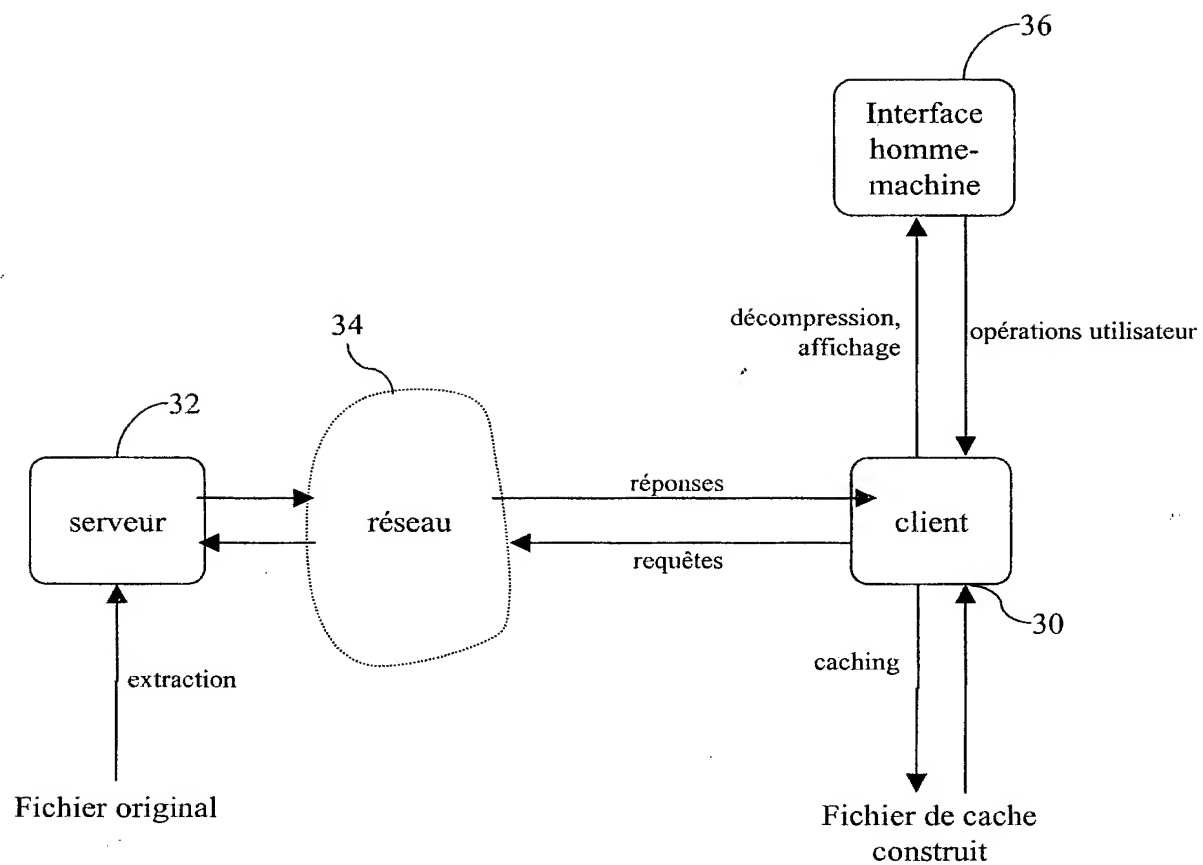
15 42. Moyen de stockage d'informations amovible, partiellement ou totalement lisible par un ordinateur ou un microprocesseur comportant des instructions de code d'un programme d'ordinateur pour l'exécution des étapes du procédé de traitement de données selon l'une des revendications 12 à 22.

20 43. Programme d'ordinateur chargeable dans un appareil programmable, caractérisé en ce qu'il comporte des séquences d'instructions ou des portions de code logiciel pour mettre en œuvre les étapes du procédé de traitement d'une requête selon l'une des revendications 1 à 11, lorsque ce programme d'ordinateur est chargé et exécuté par l'appareil programmable.

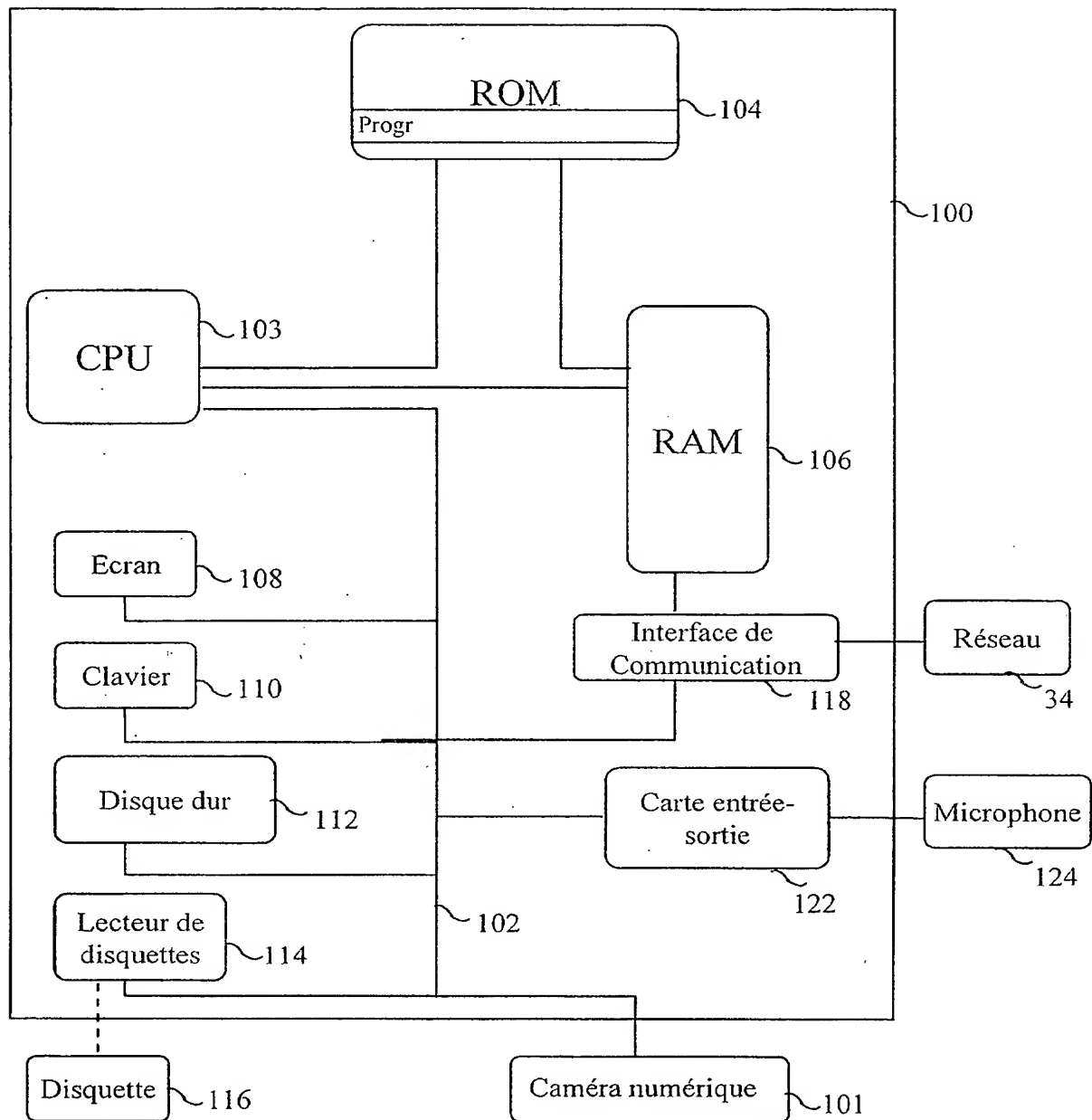
25 44. Programme d'ordinateur chargeable dans un appareil programmable, caractérisé en ce qu'il comporte des séquences d'instructions ou des portions de code logiciel pour mettre en œuvre les étapes du procédé de traitement de données selon l'une des revendications 12 à 22, lorsque ce programme d'ordinateur est chargé et exécuté par l'appareil programmable.

1/18

*Fig. 1*

*Fig. 1*

2/18

*Fig. 2*

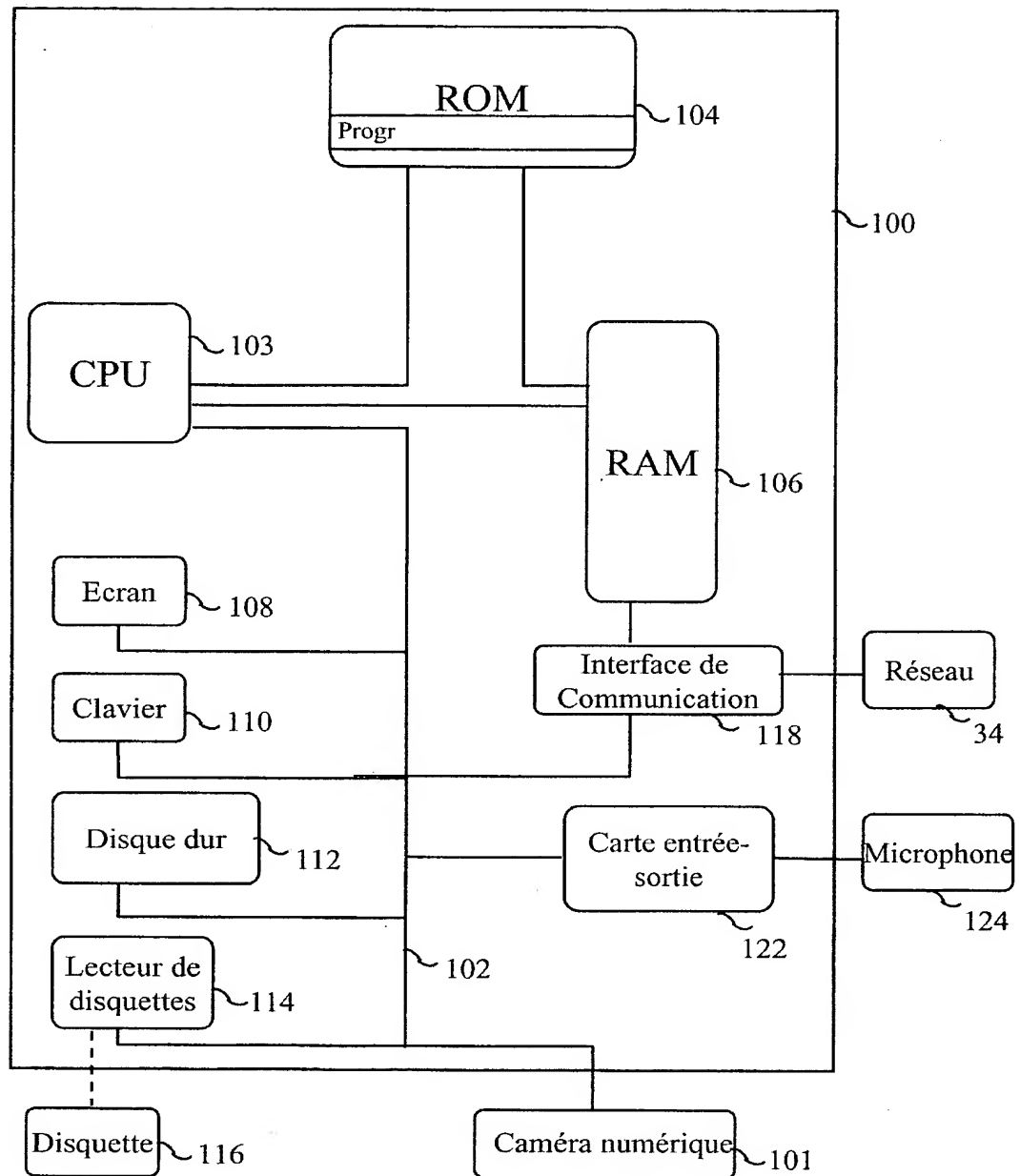


Fig. 2

3/18

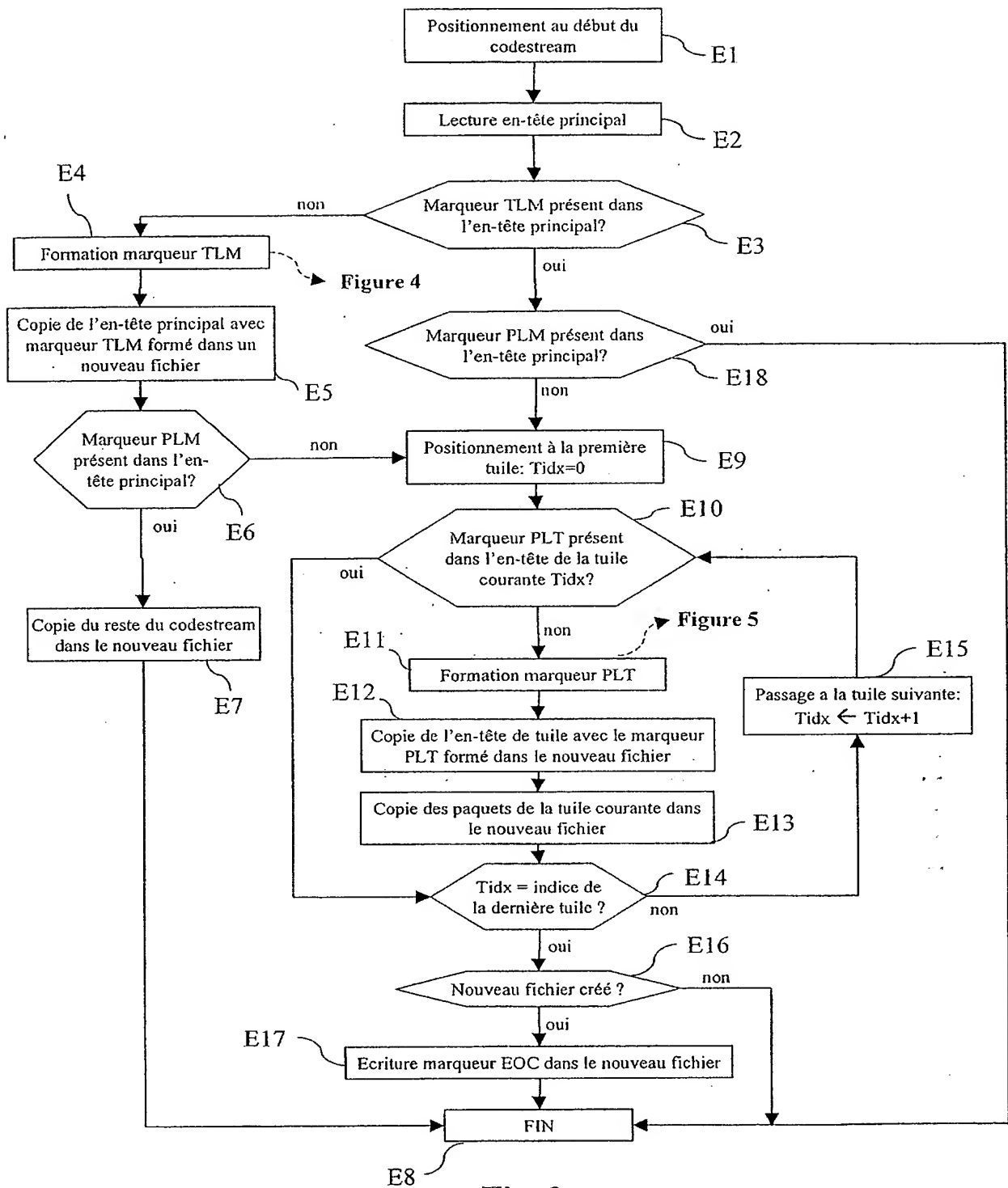


Fig. 3

3/18

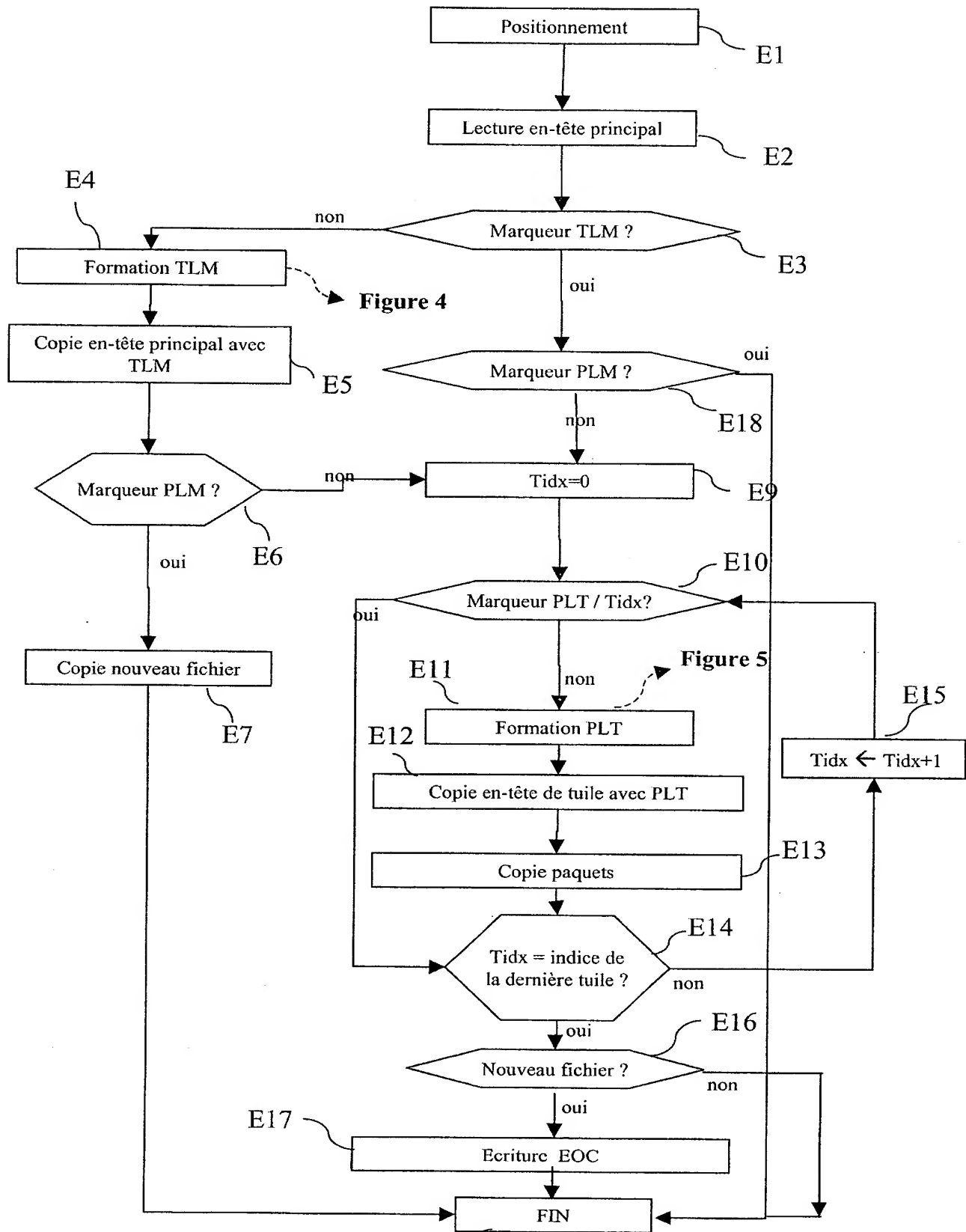
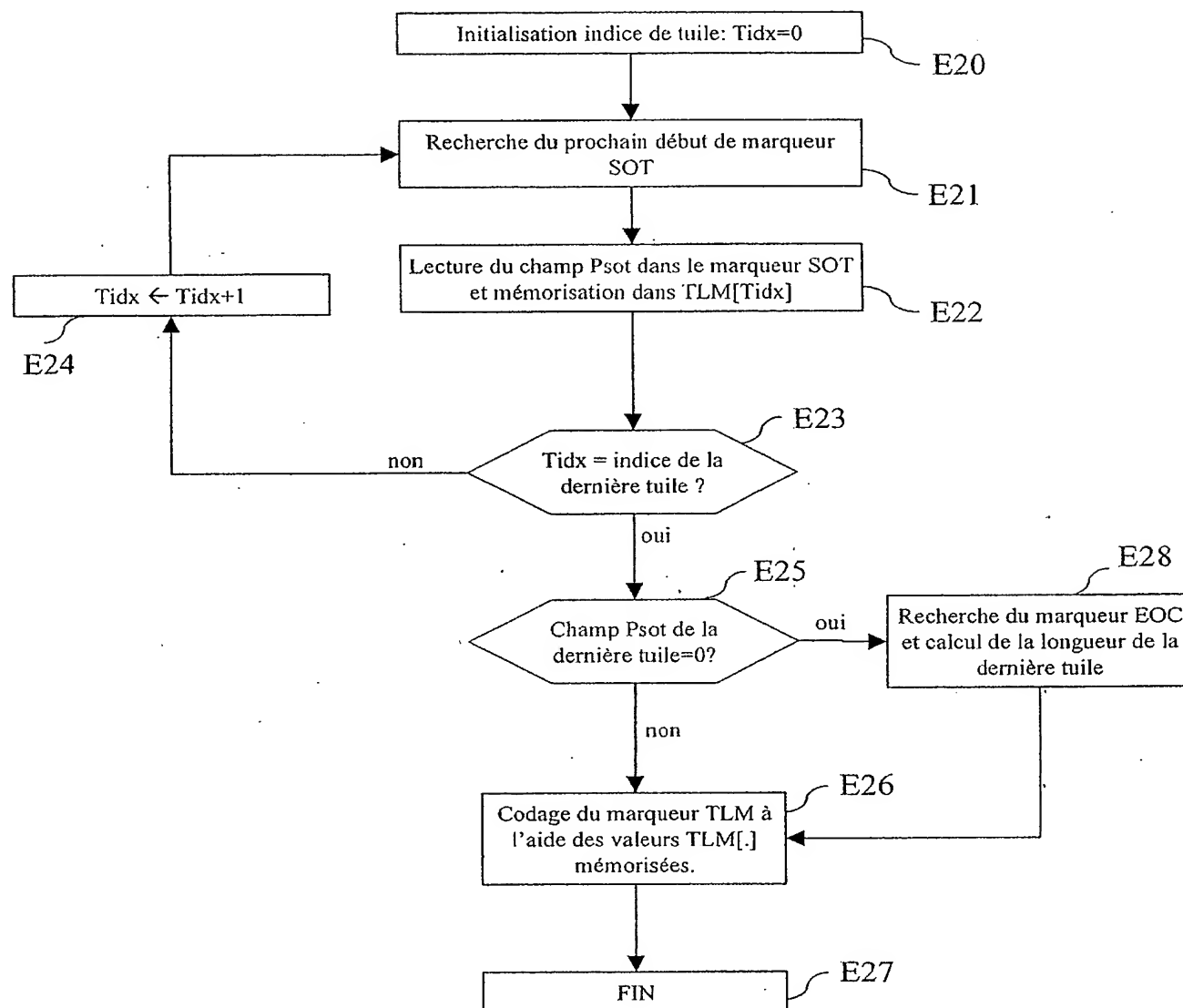
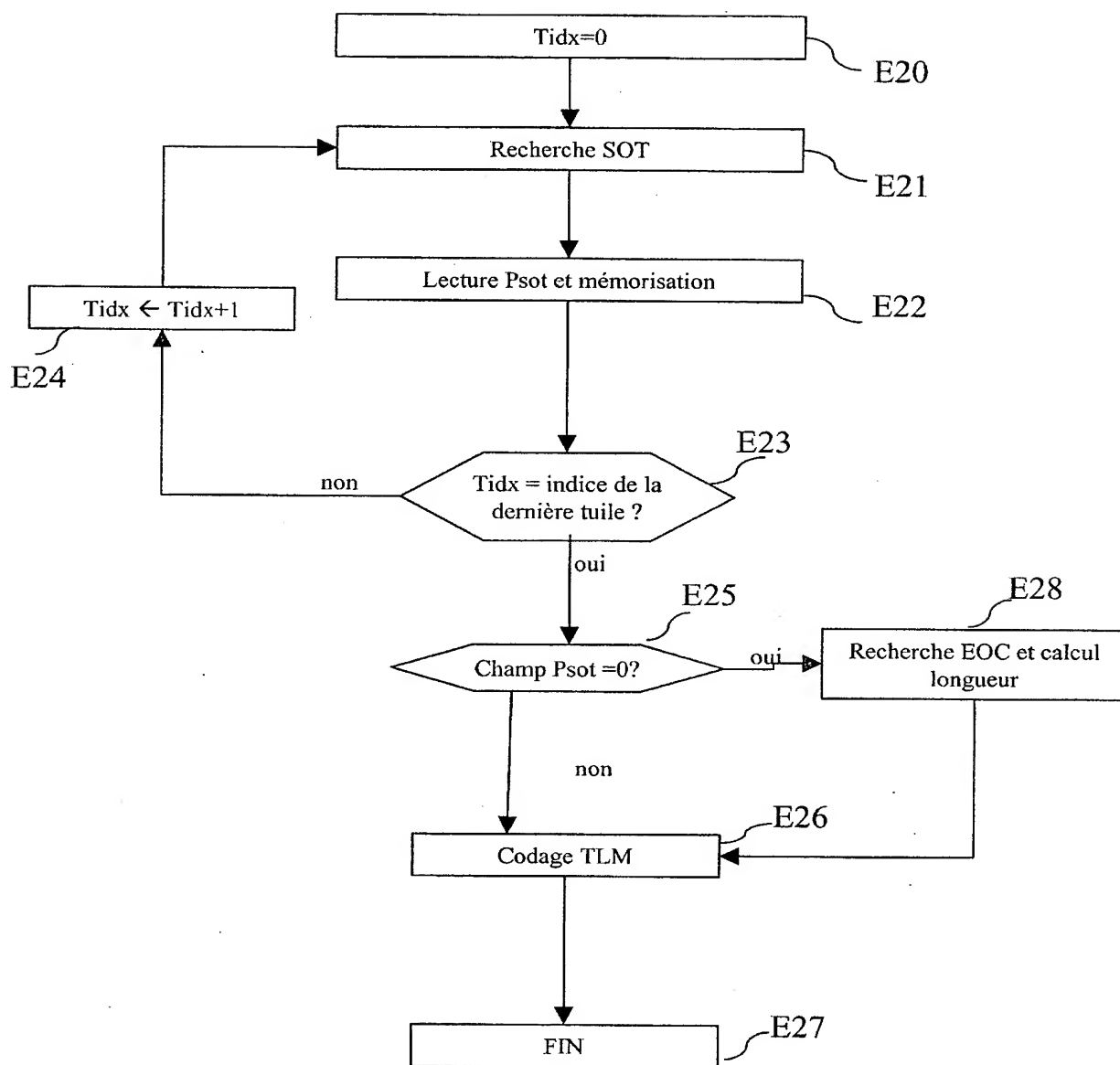


Fig. 3

4/18

*Fig. 4*

*Fig. 4*

5/18

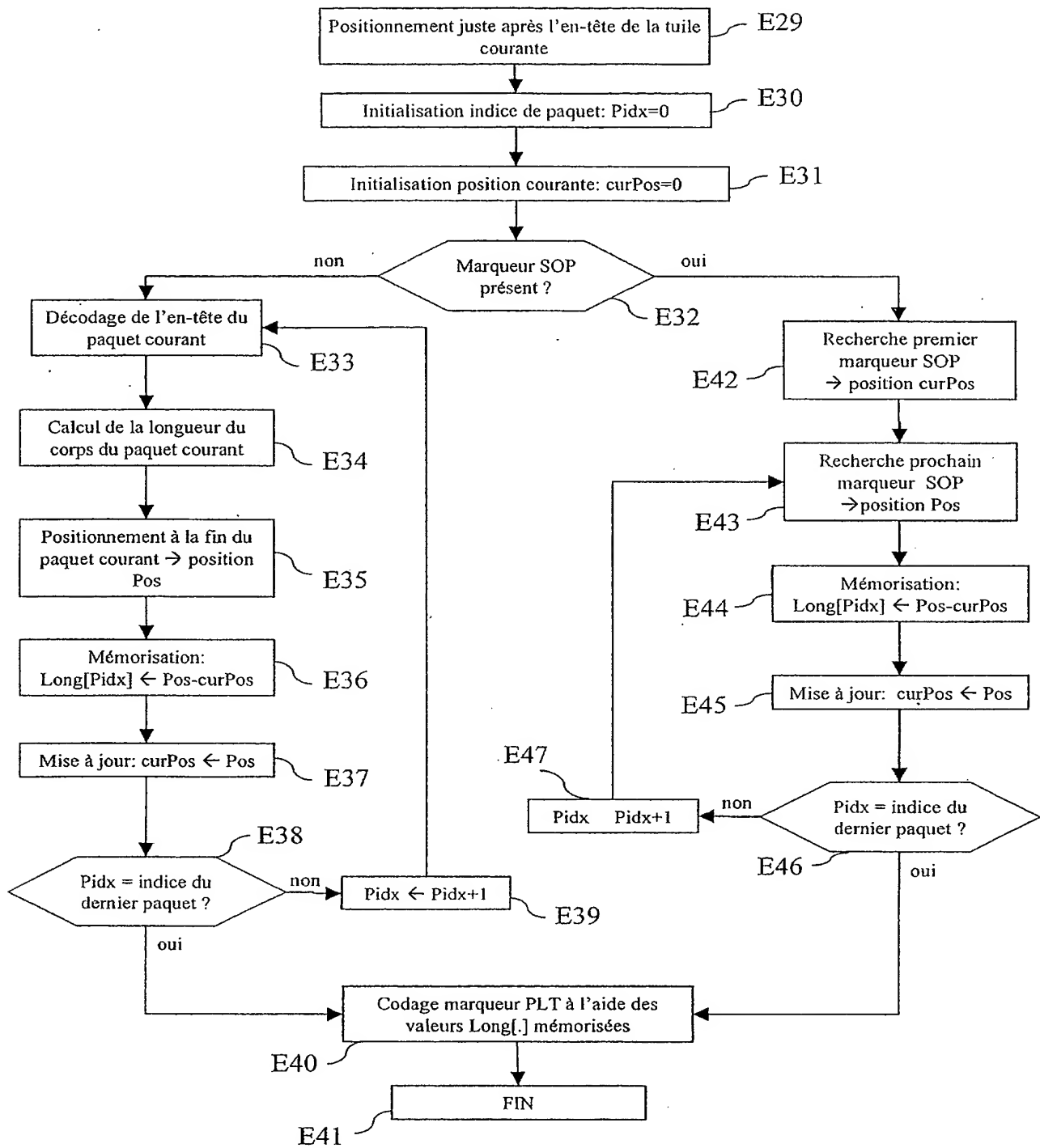


Fig. 5

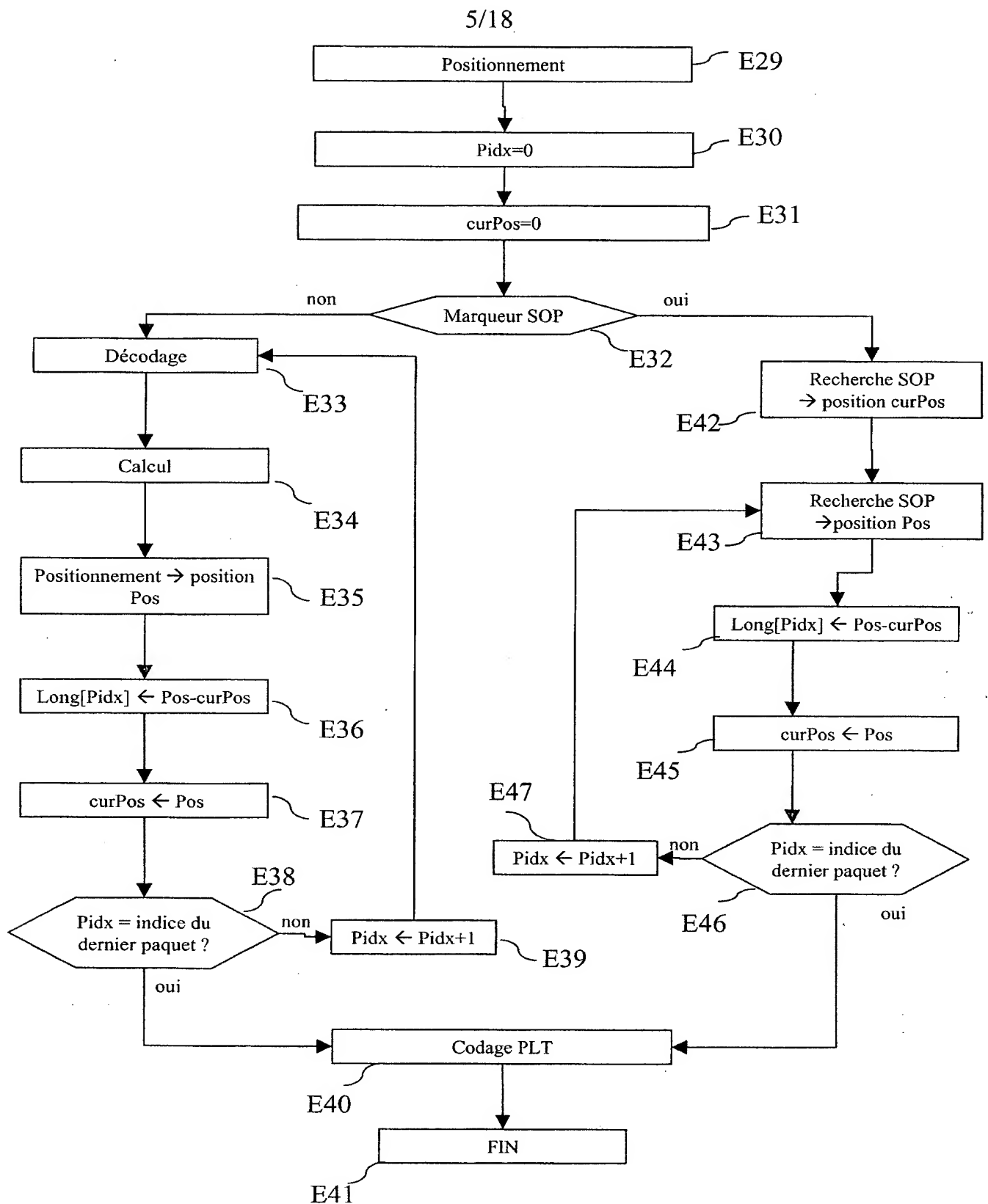


Fig. 5

6/18

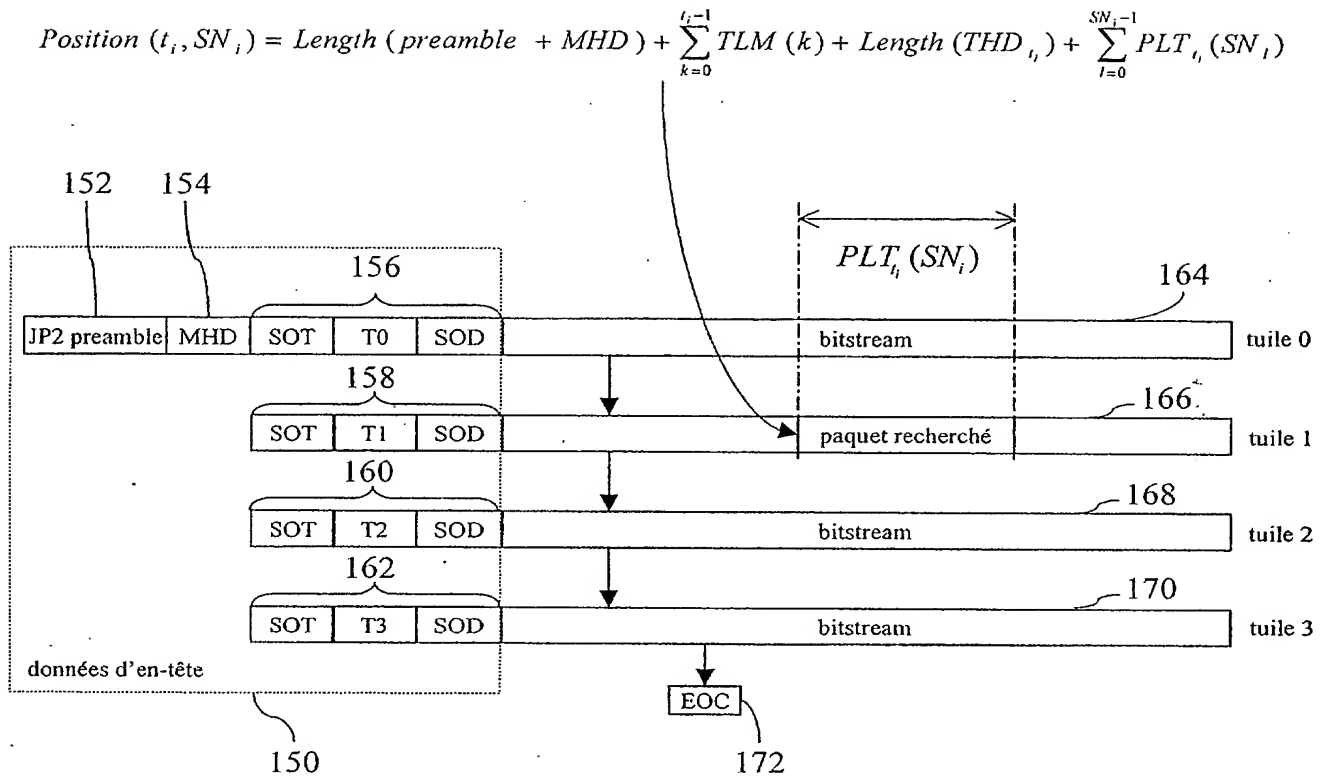


Fig. 6

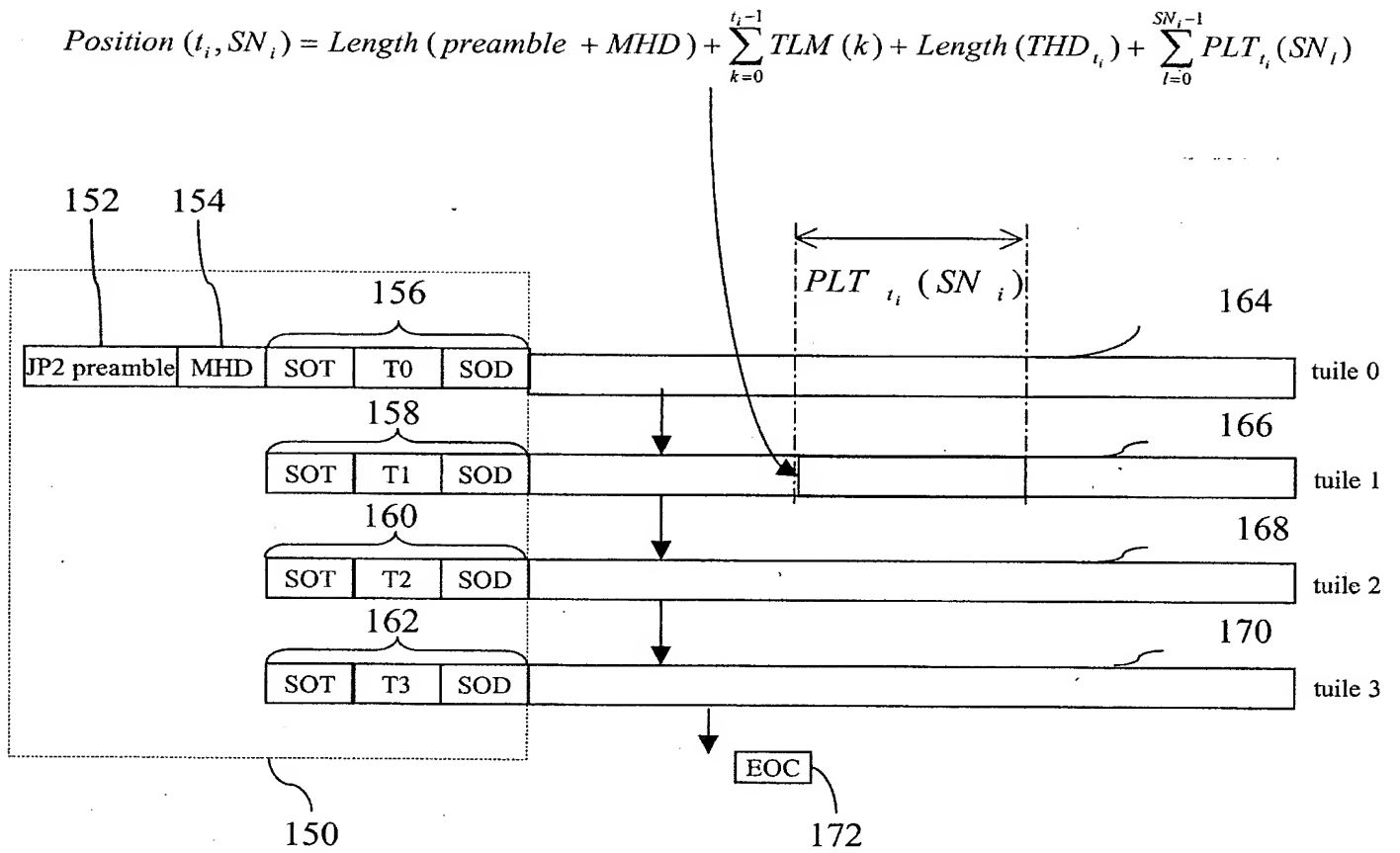
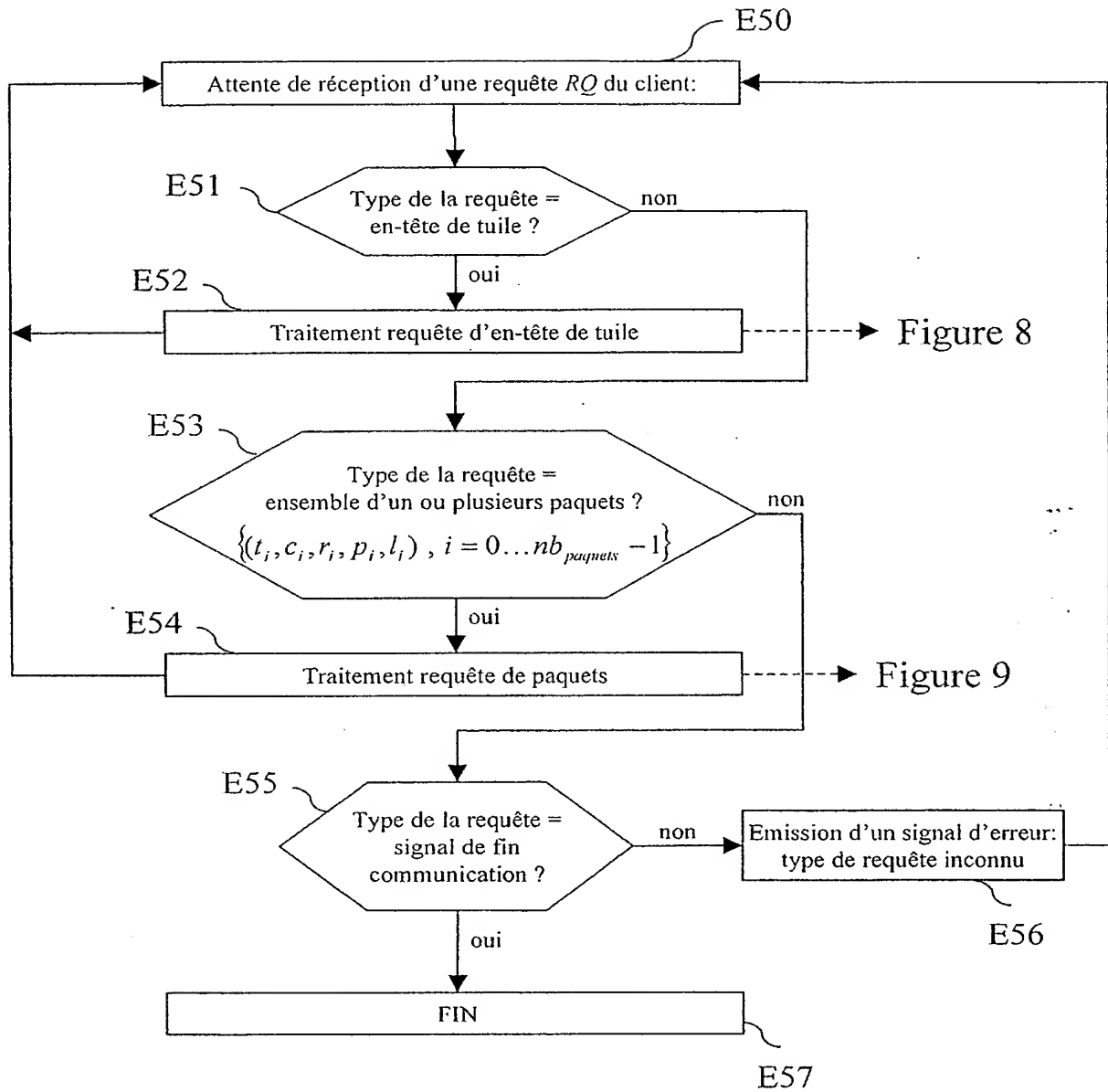
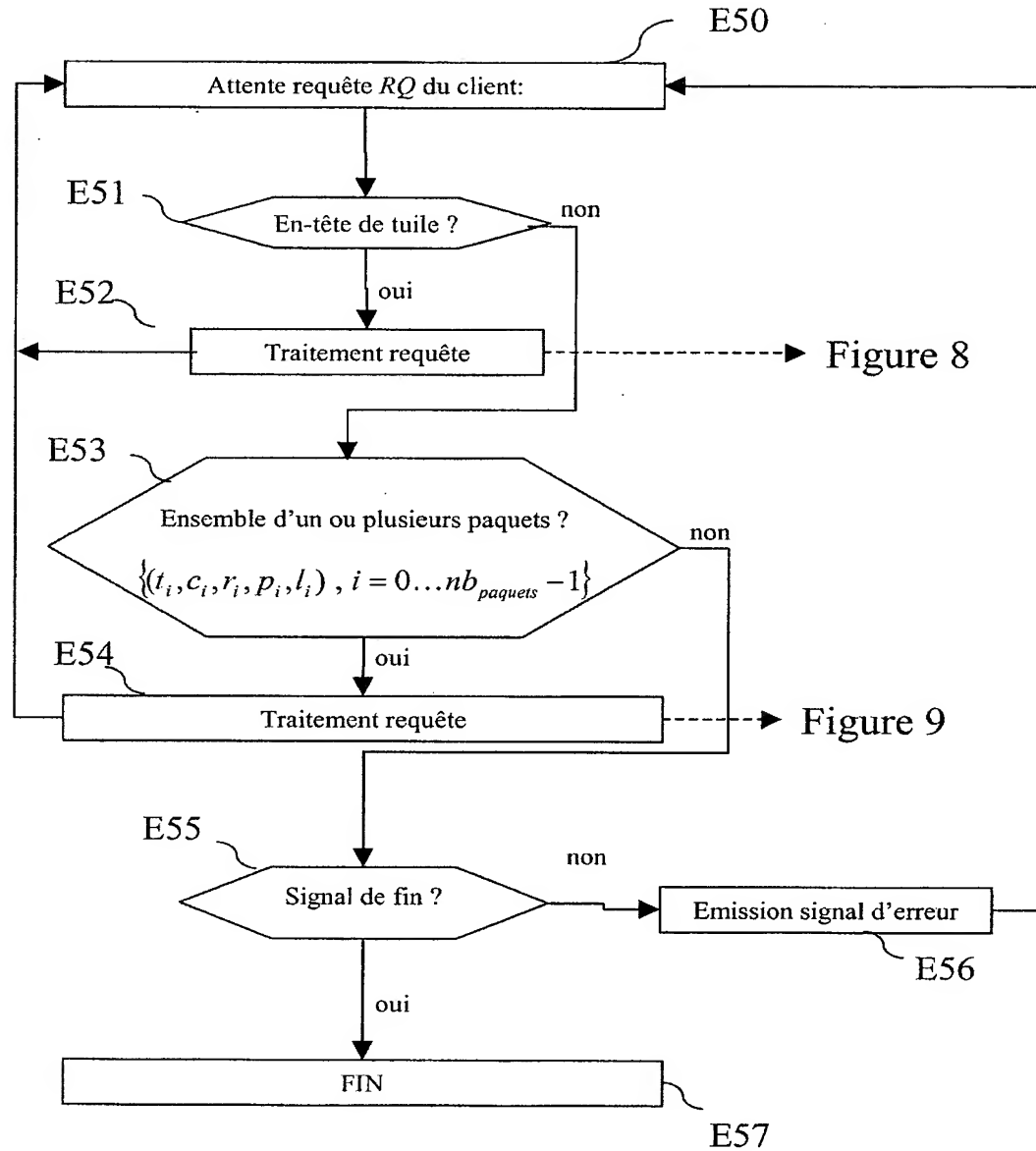


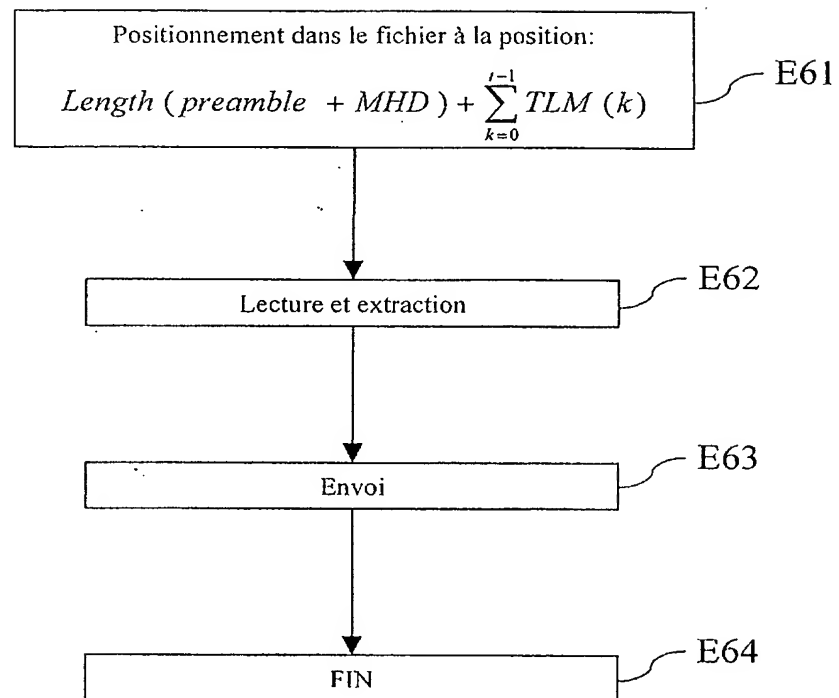
Fig. 6

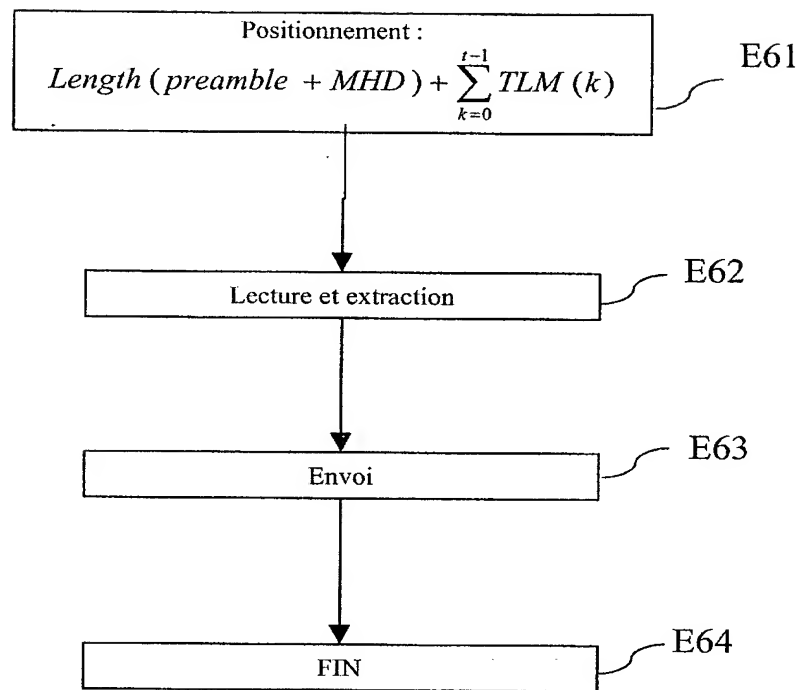
7/18

*Fig. 7*

**Fig. 7**

8/18

*Fig. 8*

**Fig. 8**

9/18

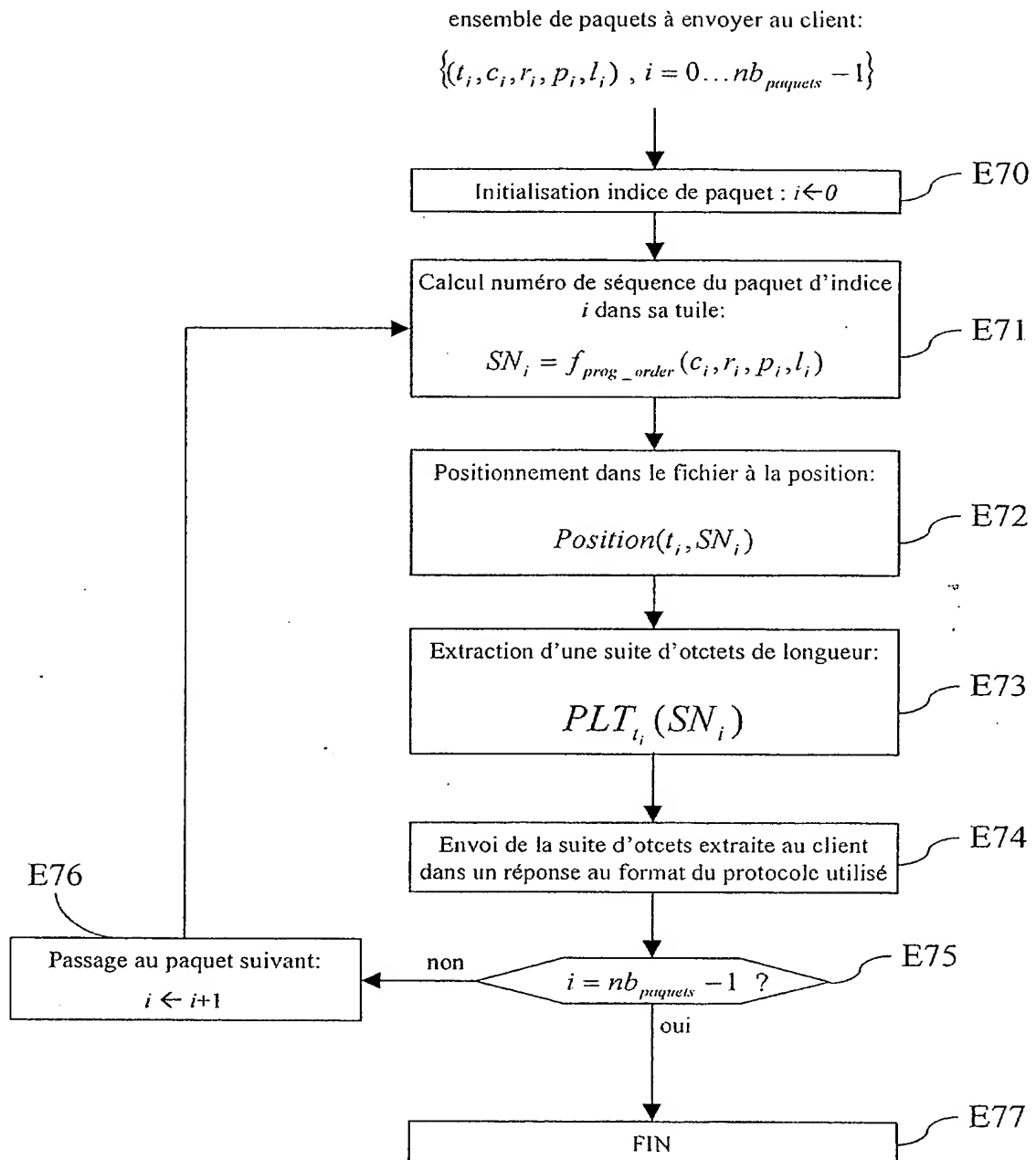
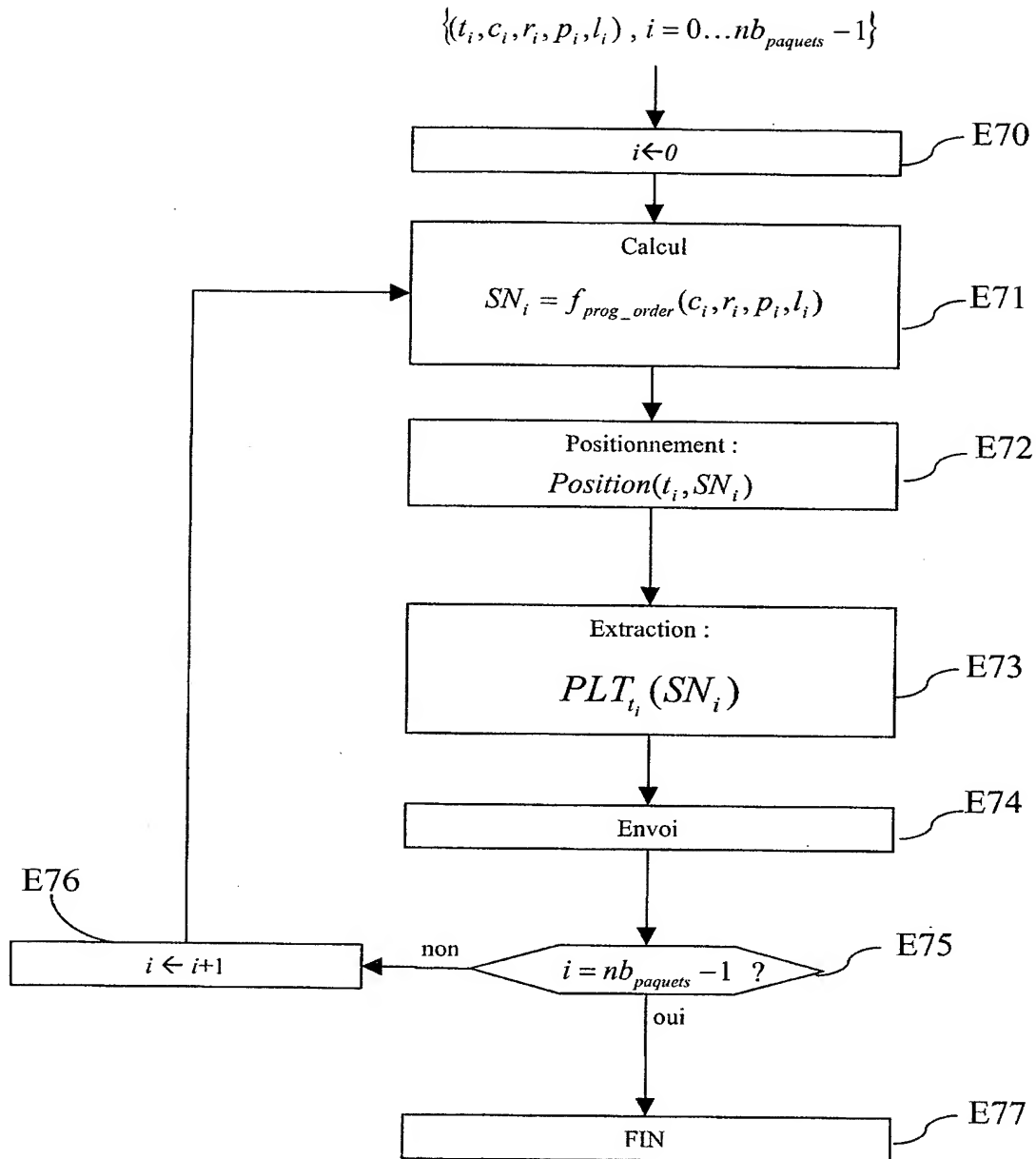


Fig. 9

9/18

**Fig. 9**

10/18

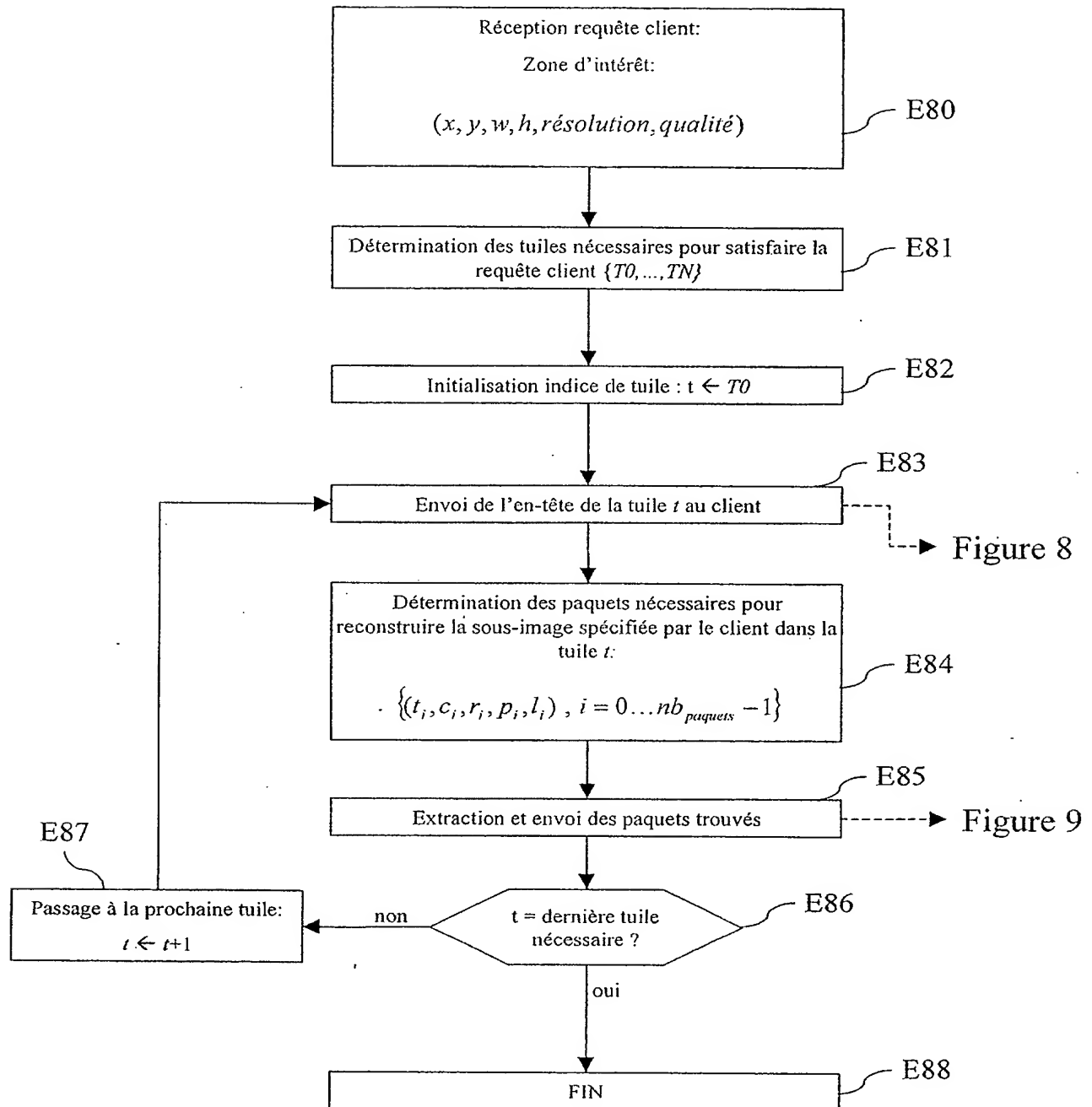


Fig. 10

10/18

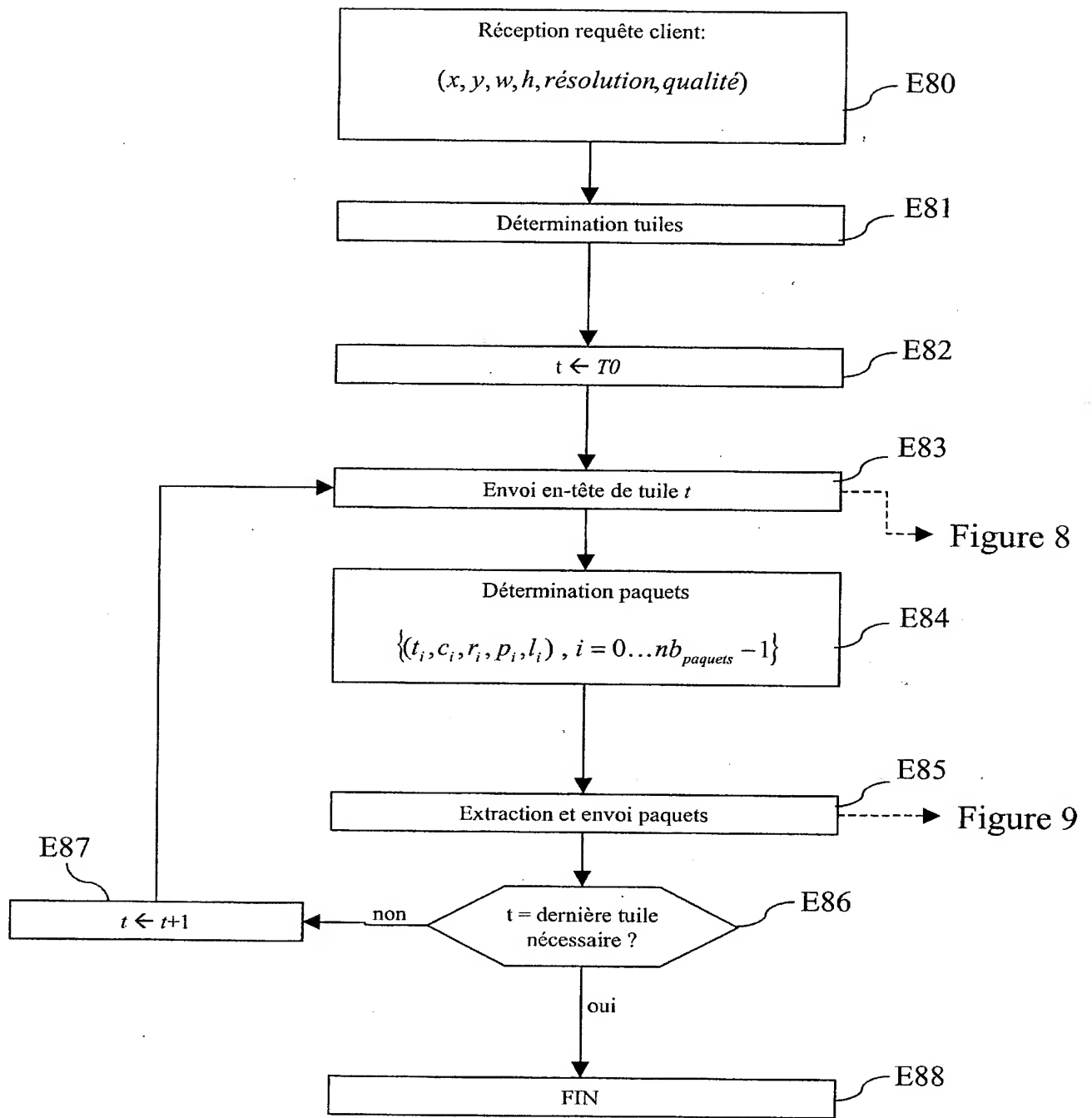


Fig. 10

11/18

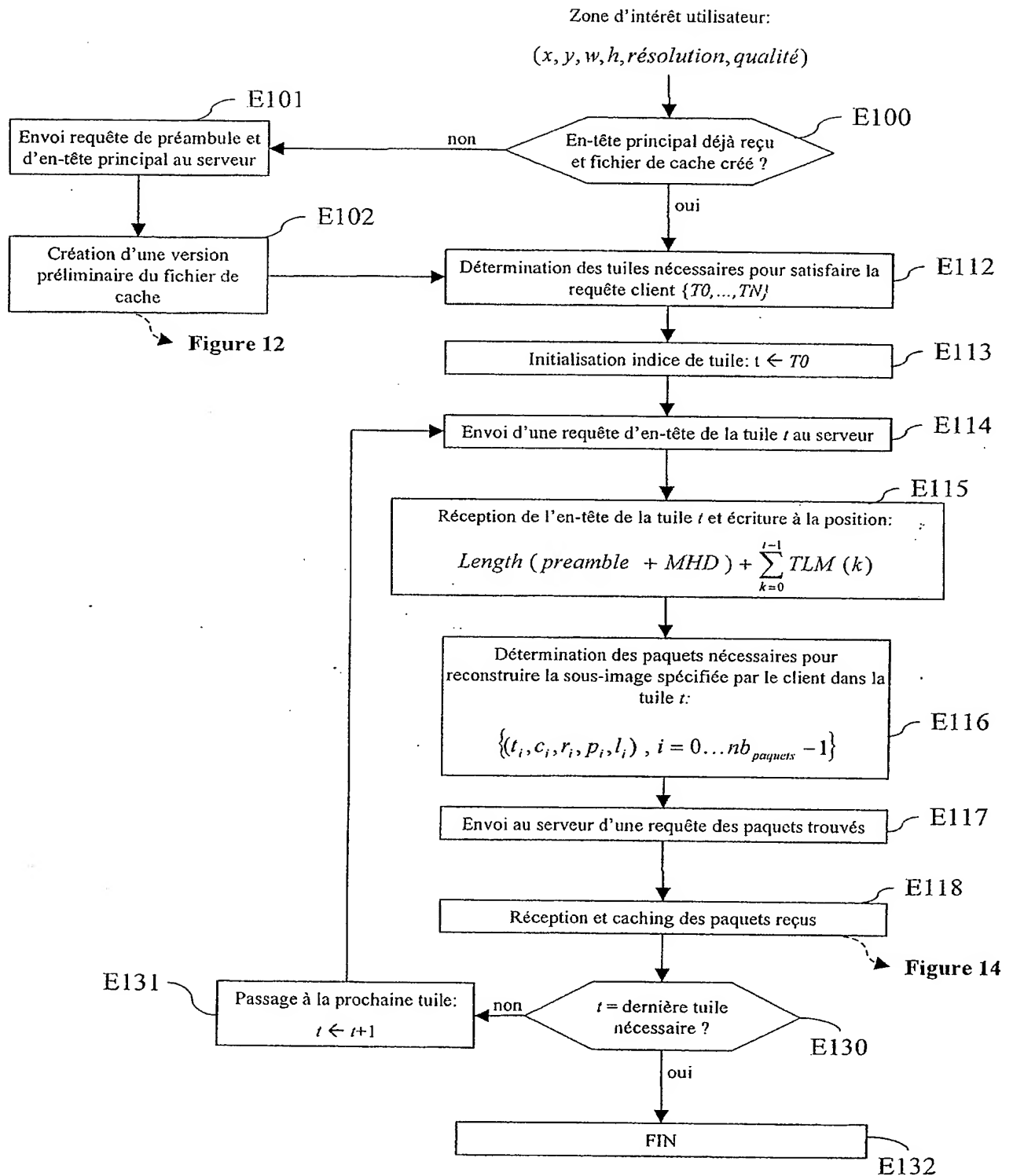
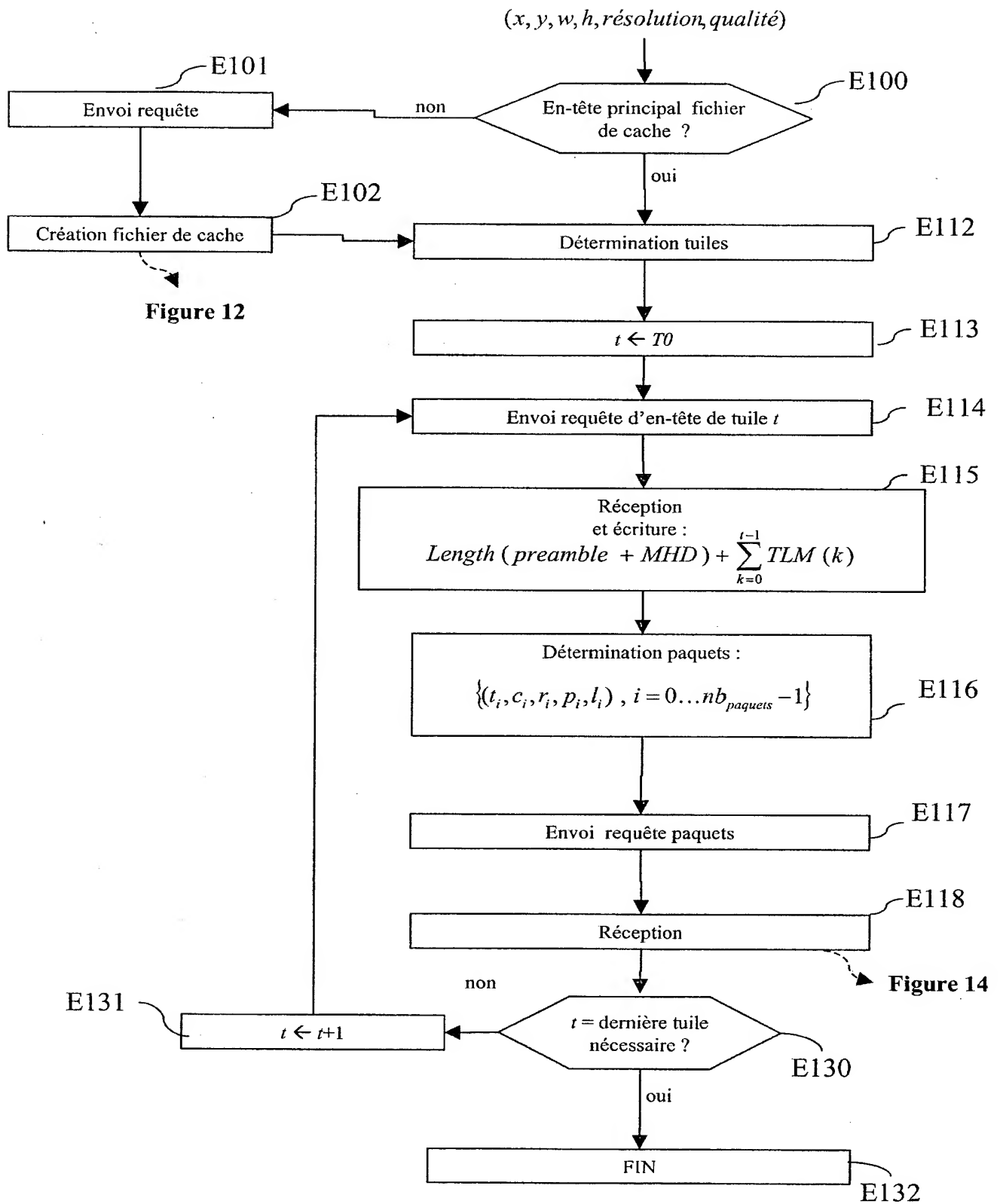
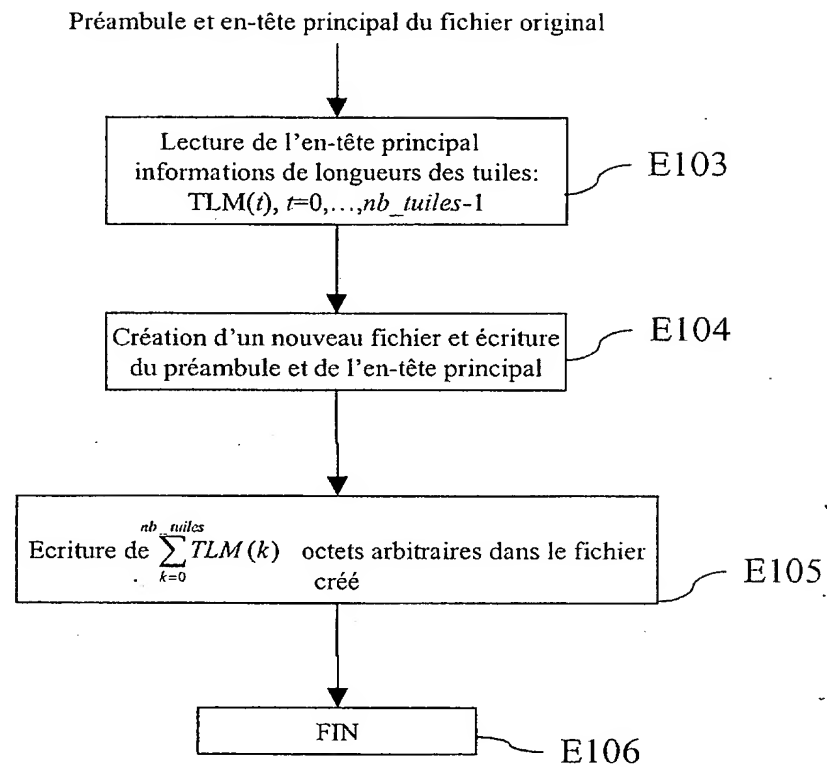


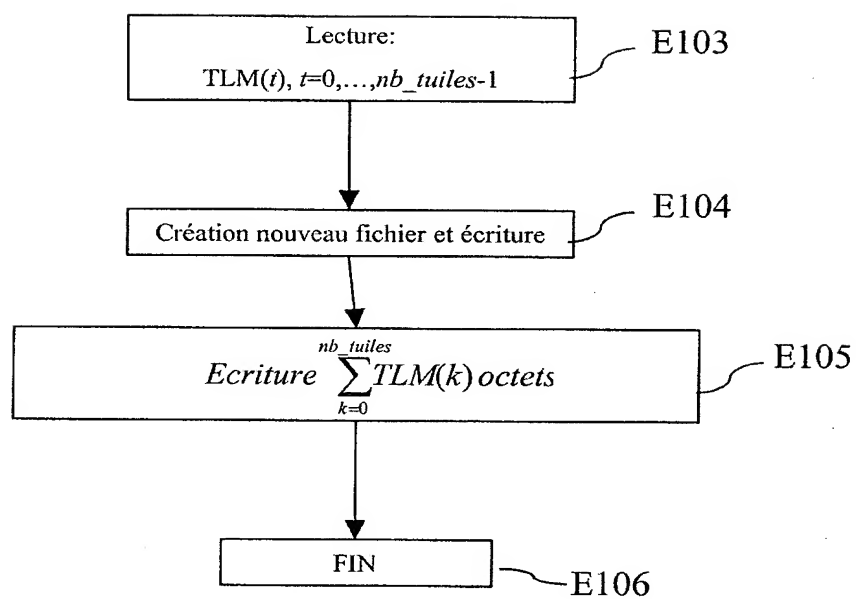
Fig. 11

11/18

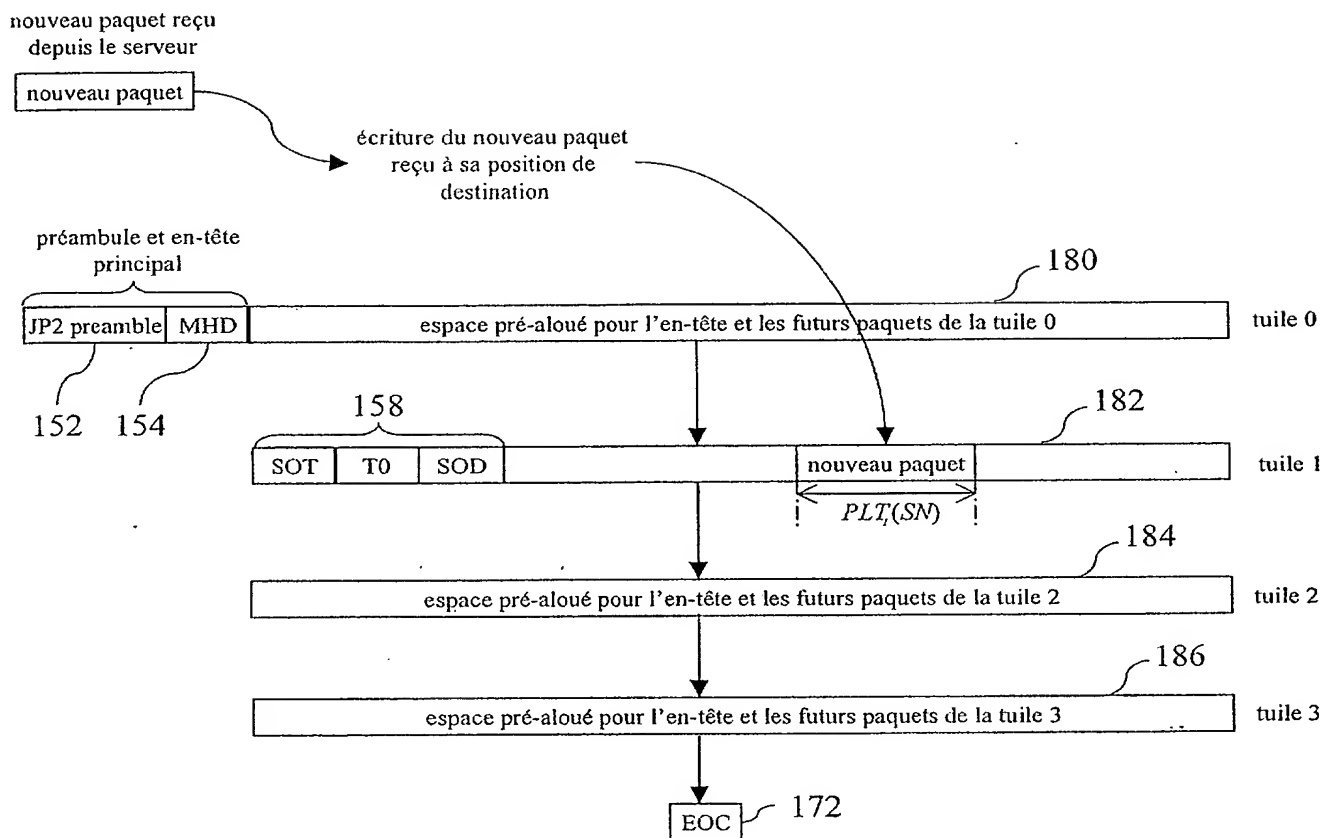


12/18

*Fig. 12*

**Fig. 12**

13/18

*Fig. 13*

13/18

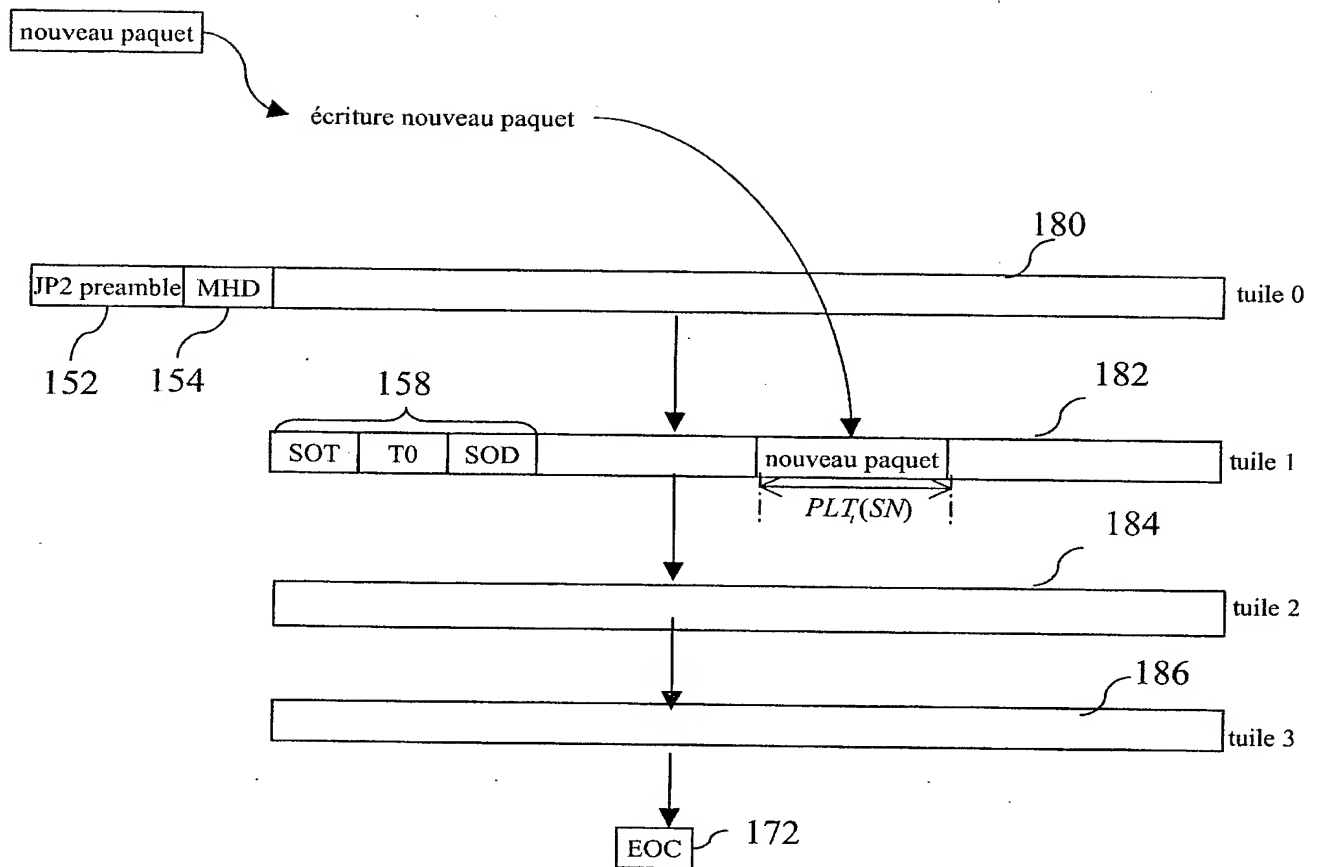
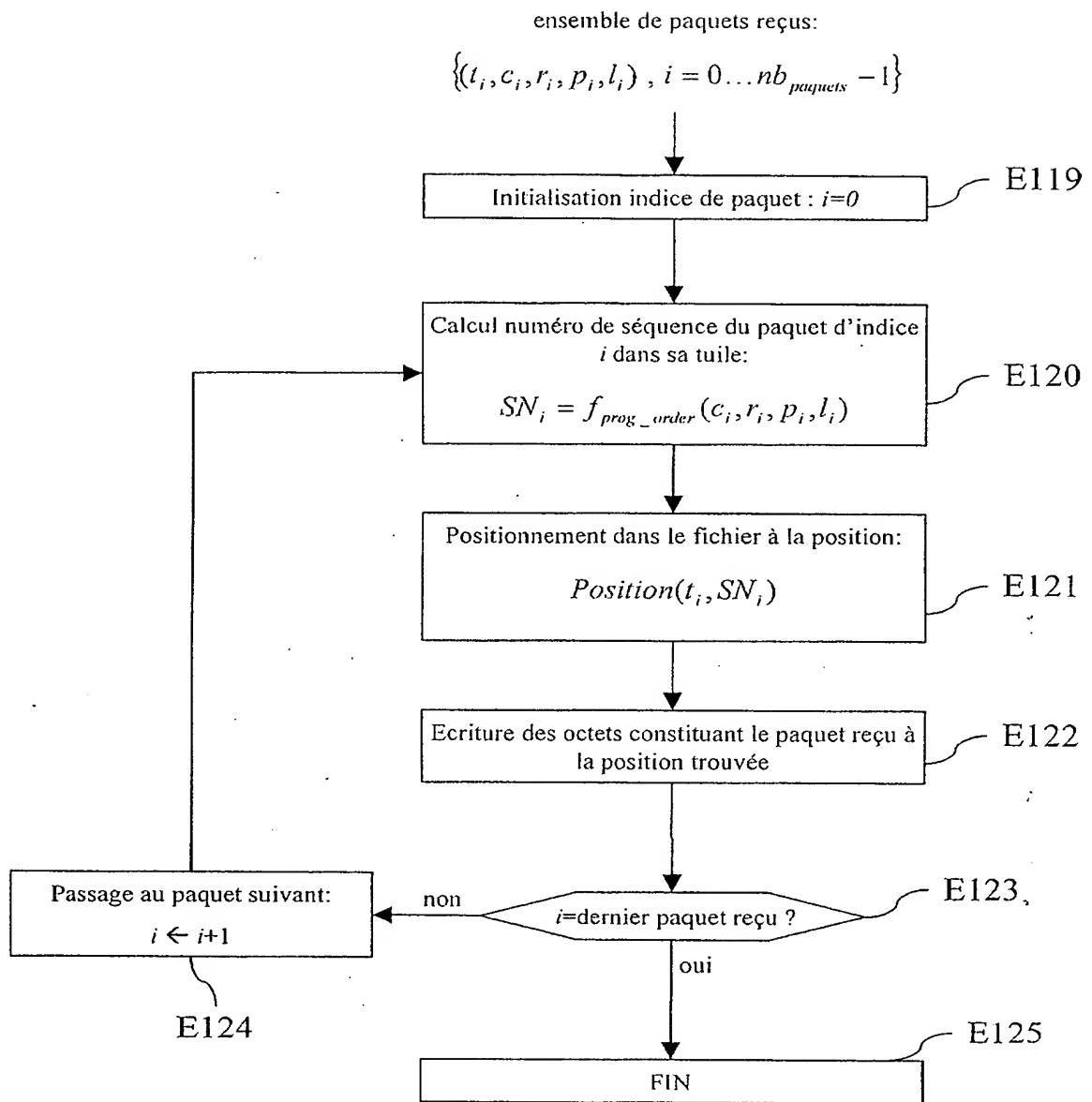
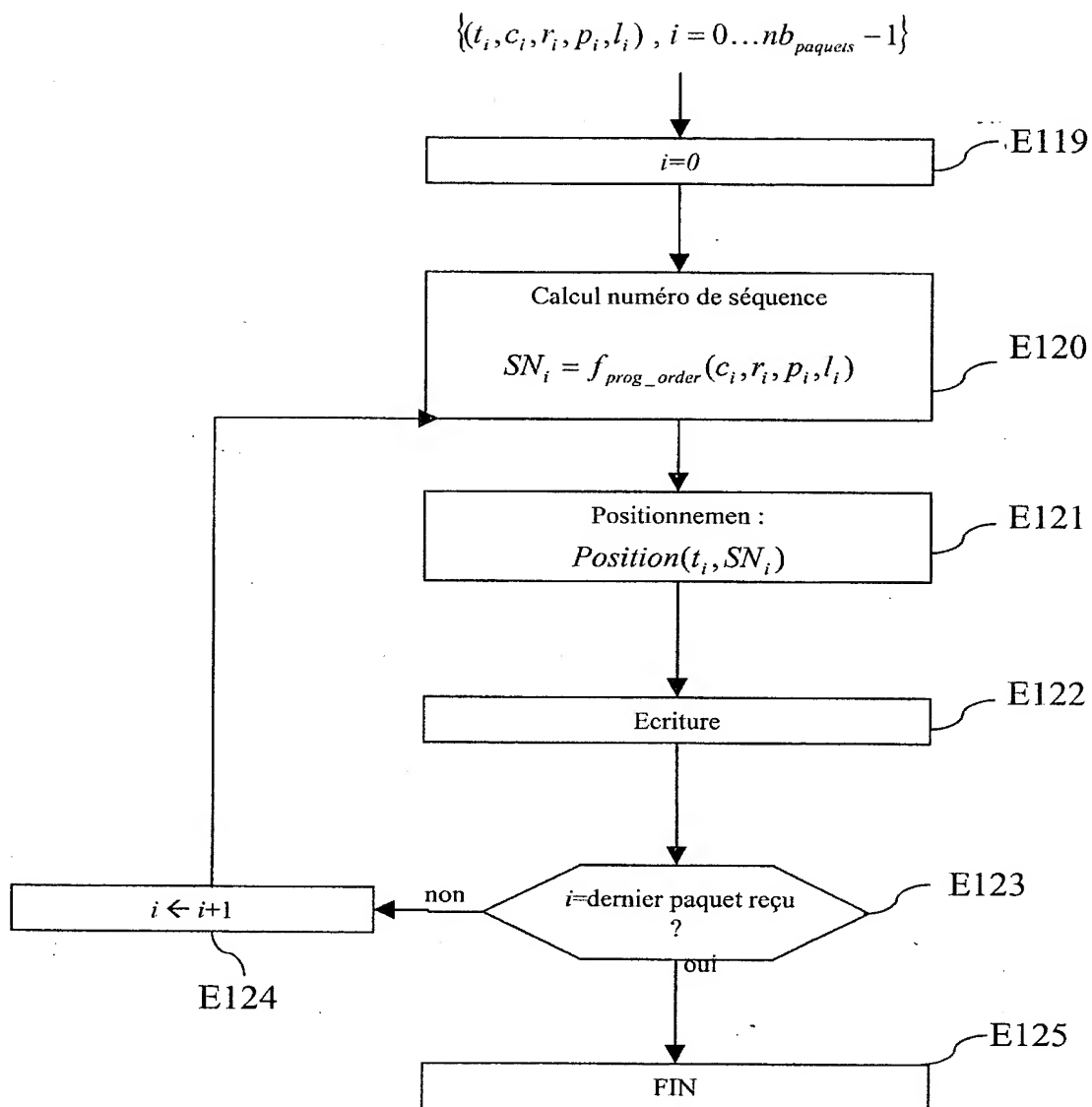


Fig. 13

14/18

*Fig. 14*

**Fig. 14**

15/18

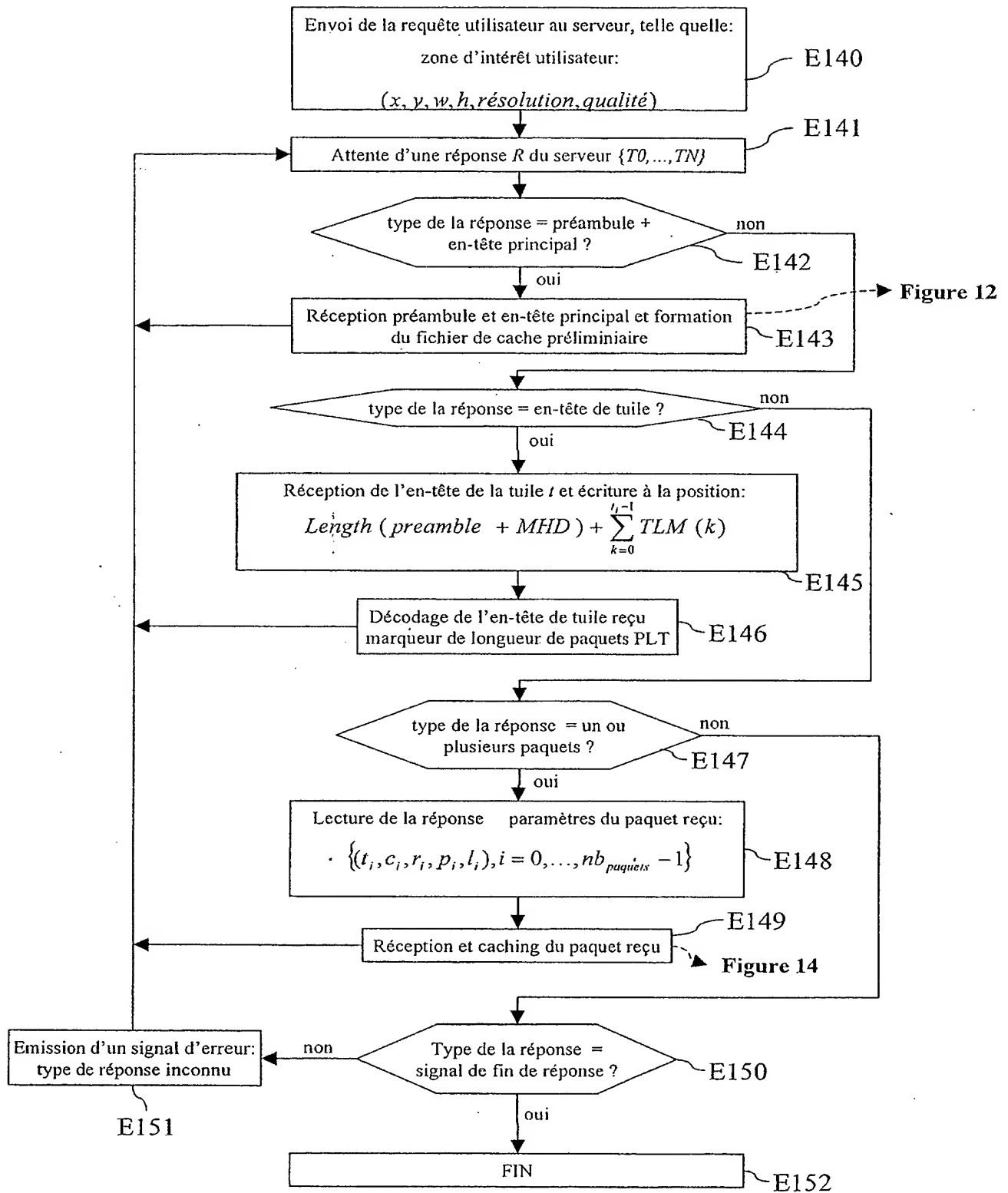


Fig. 15

15/18

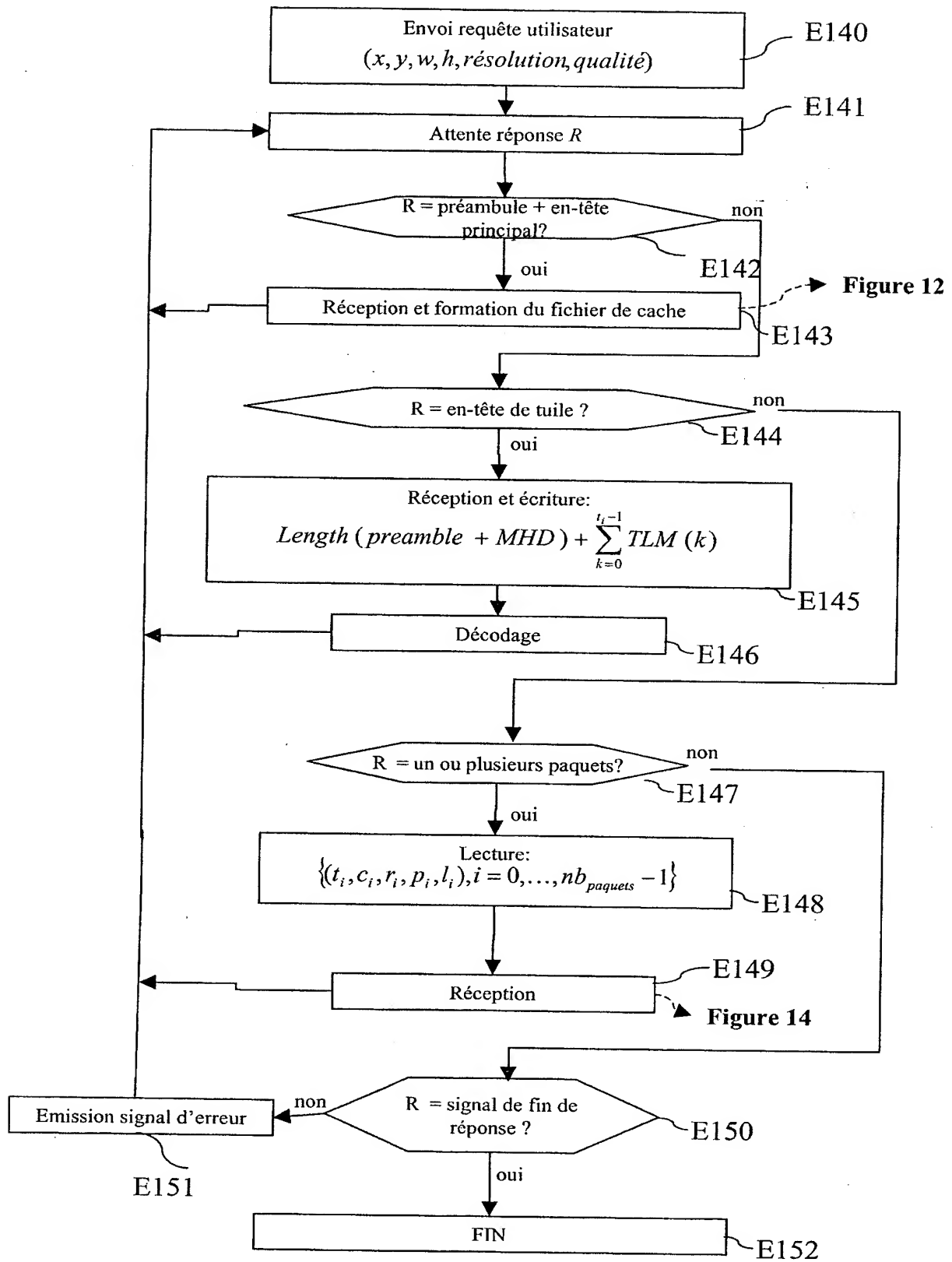


Fig. 15

16/18

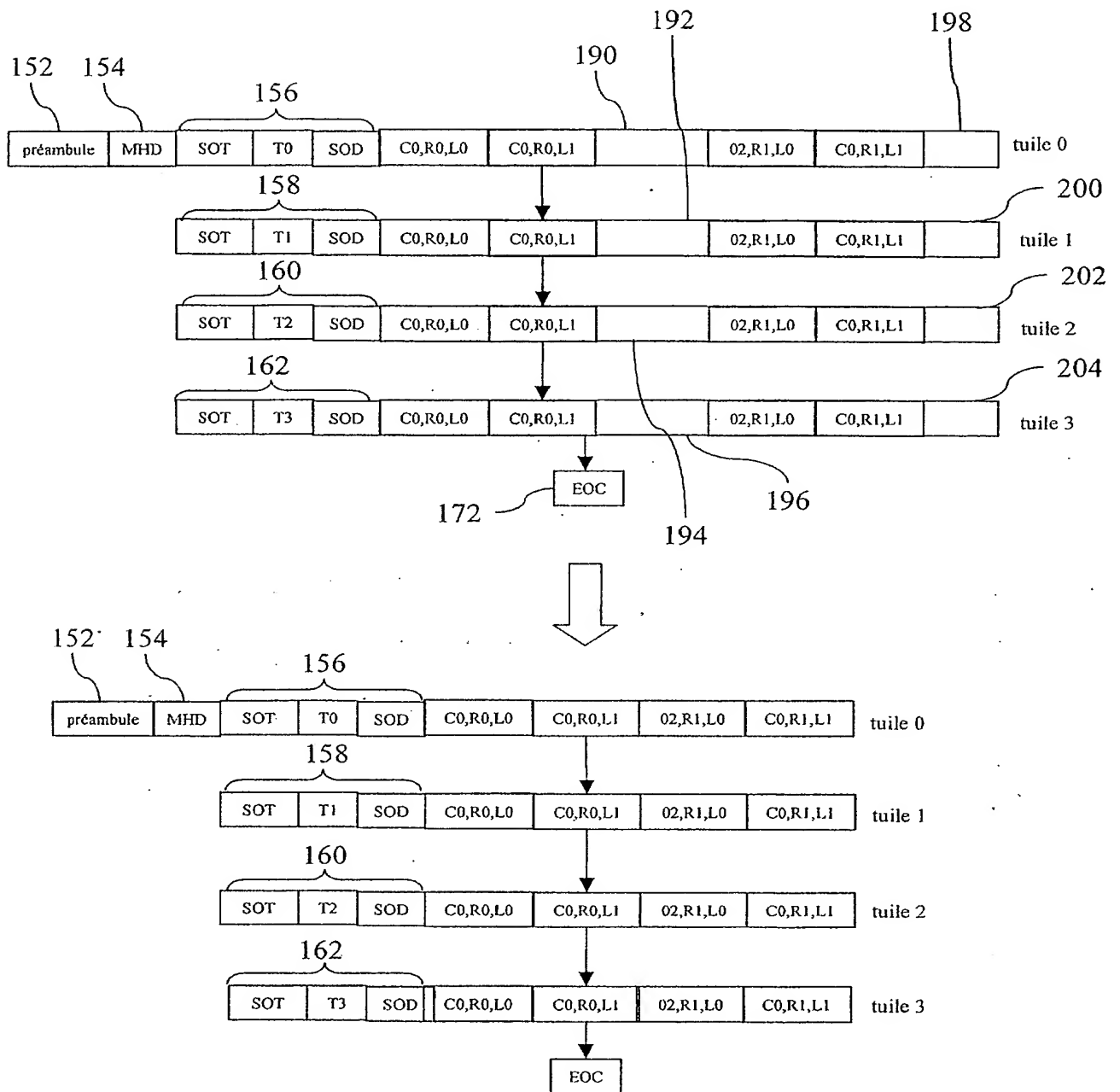


Fig. 16

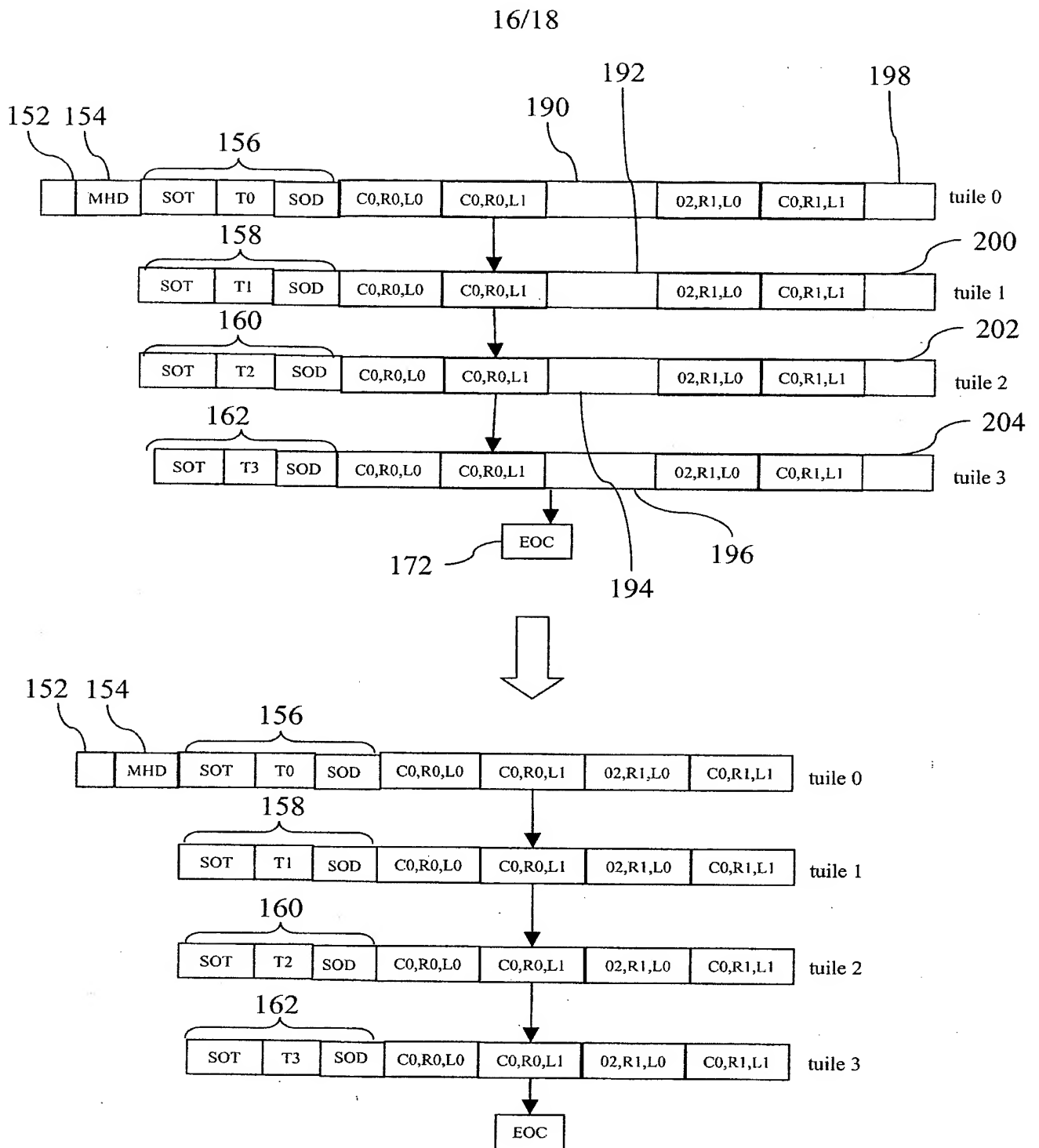


Fig. 16

17/18

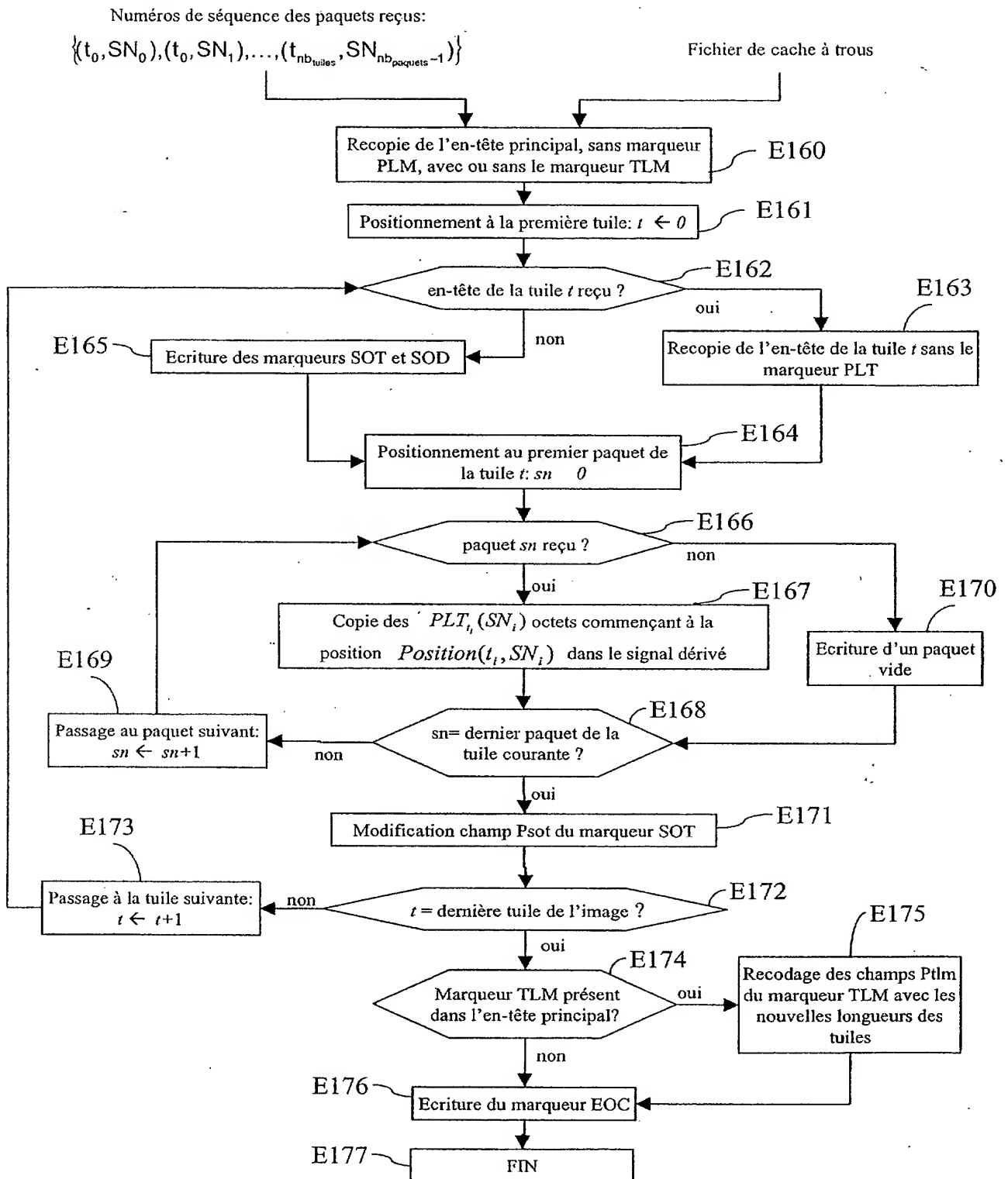


Fig. 17

17/18

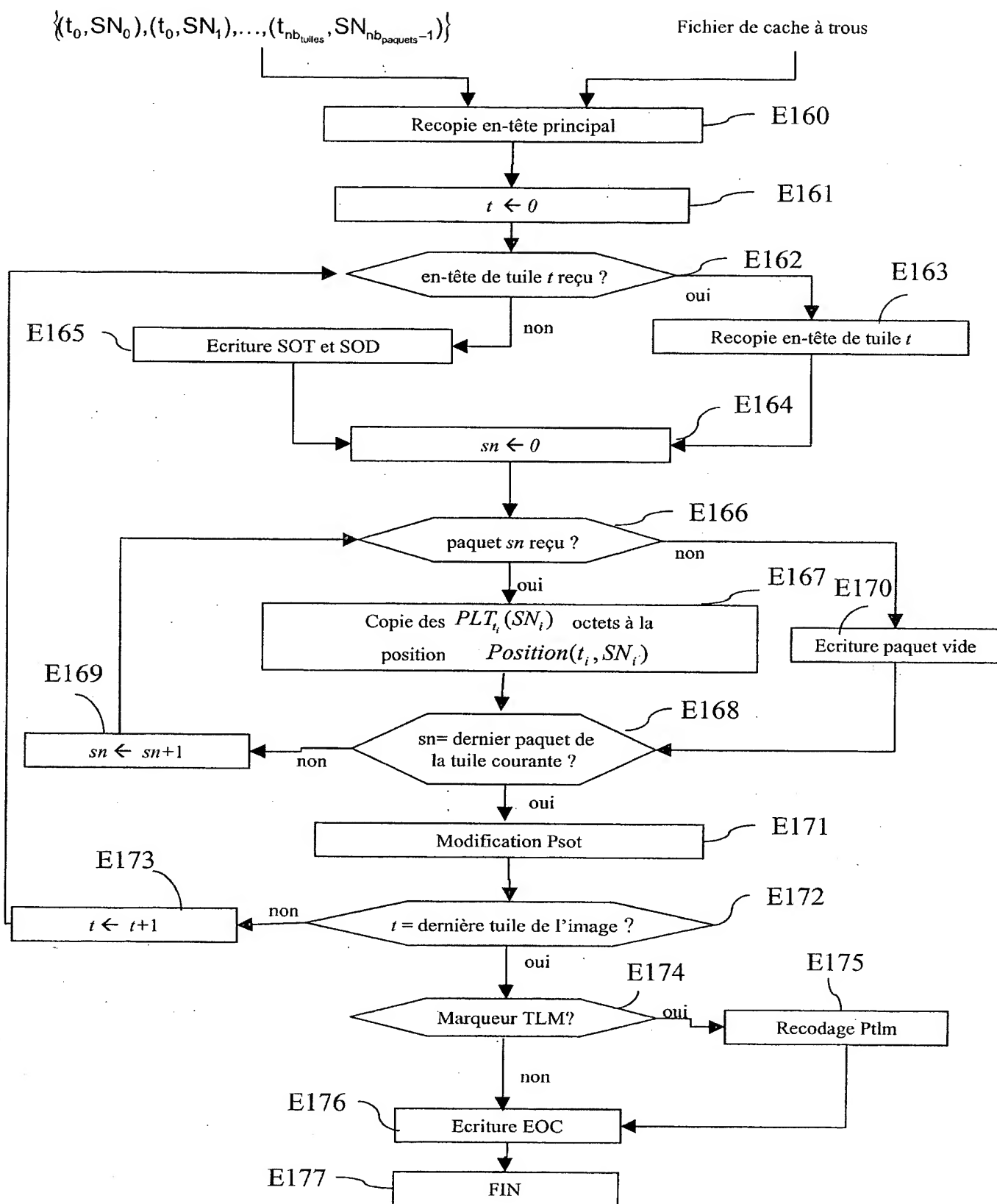


Fig. 17

18/18

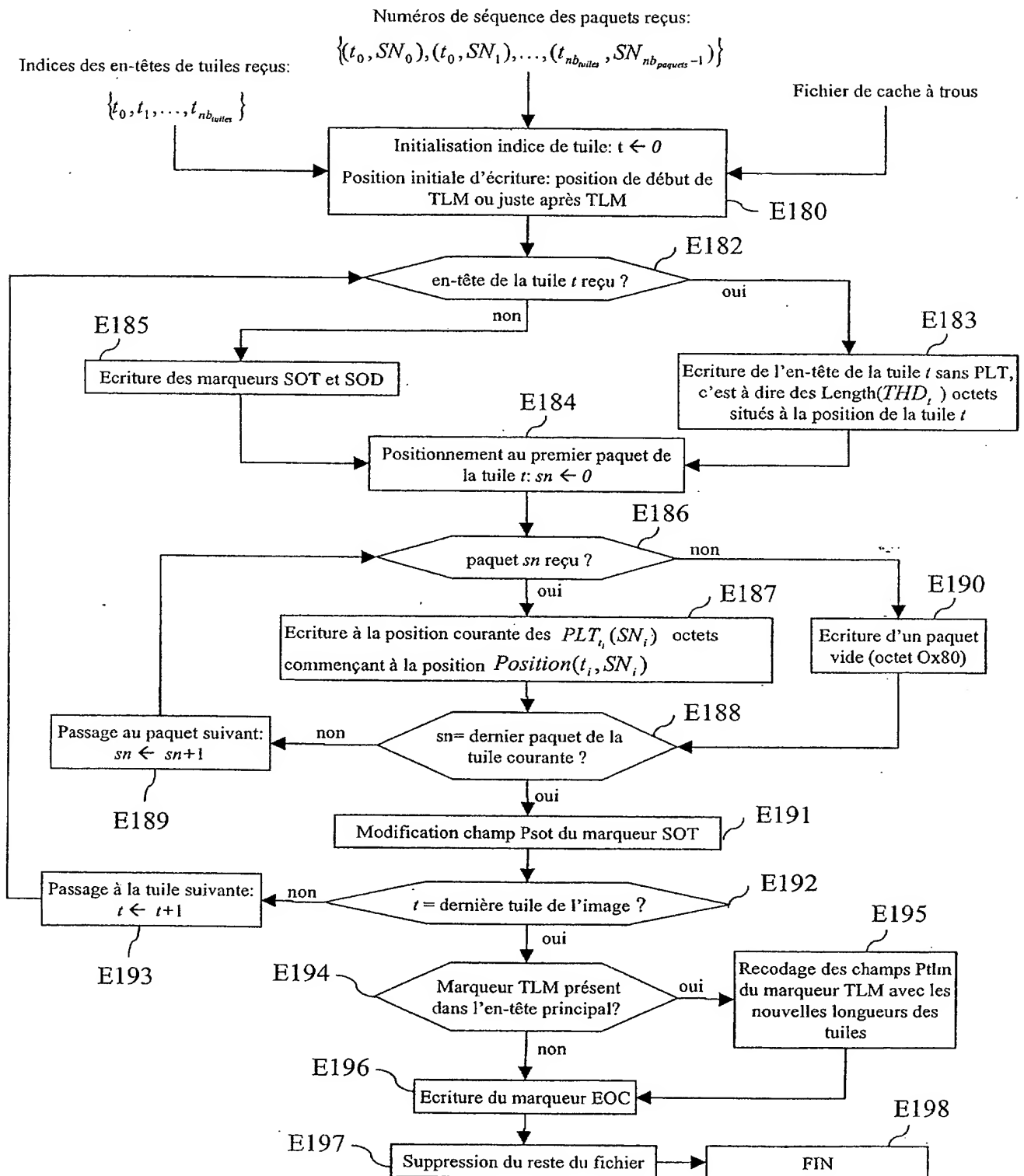


Fig. 18

18/18

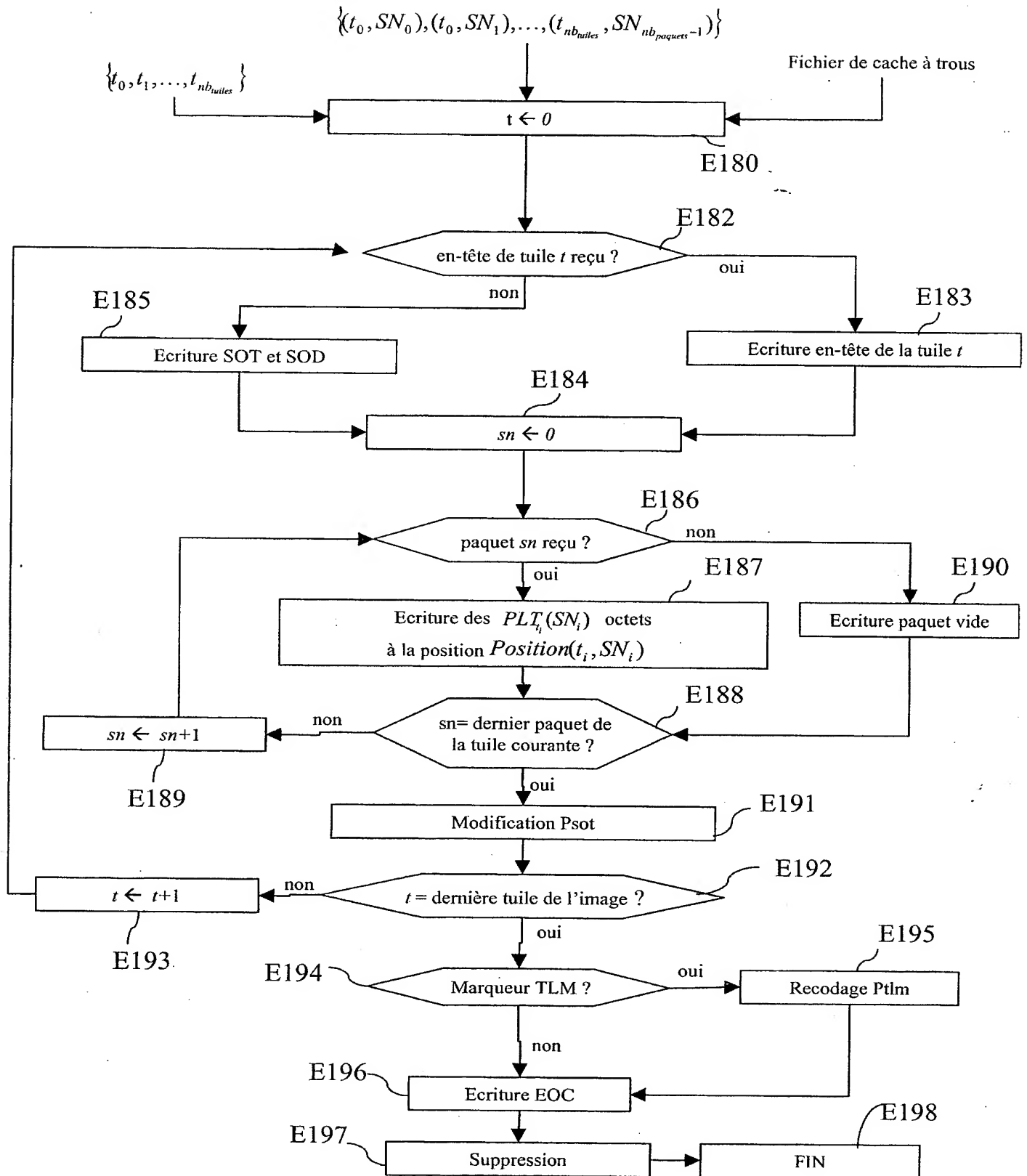


Fig. 18

DÉPARTEMENT DES BREVETS

26 bis, rue de Saint Pétersbourg
75800 Paris Cedex 08

Téléphone : 33 (1) 53 04 53 04 Télécopie : 33 (1) 42 94 86 54

DÉSIGNATION D'INVENTEUR(S) Page N° 1.11.

(Si le demandeur n'est pas l'inventeur ou l'unique inventeur)



Cet imprimé est à remplir lisiblement à l'encre noire

DB 113 W / 300301

Vos références pour ce dossier (facultatif)		BIF023183/MP/MPA	
N° D'ENREGISTREMENT NATIONAL		020 8900	
TITRE DE L'INVENTION (200 caractères ou espaces maximum)			
Procédé et dispositif de traitement d'une requête ou de données numériques compressées			
LE(S) DEMANDEUR(S) :			
CANON KABUSHIKI KAISHA			
DESIGNE(NT) EN TANT QU'INVENTEUR(S) : (Indiquez en haut à droite «Page N° 1/1» S'il y a plus de trois inventeurs, utilisez un formulaire identique et numérotez chaque page en indiquant le nombre total de pages).			
Nom		LE LEANNEC	
Prénoms		Fabrice	
Adresse	Rue	La Gaudais,	
	Code postal et ville	35510 CESSON SEVIGNE, France	
Société d'appartenance (facultatif)			
Nom		ONNO	
Prénoms		Patrice	
Adresse	Rue	60 avenue du Sergent Maginot,	
	Code postal et ville	35000 RENNES, France	
Société d'appartenance (facultatif)			
Nom			
Prénoms			
Adresse	Rue		
	Code postal et ville		
Société d'appartenance (facultatif)			
DATE ET SIGNATURE(S) DU (DES) DEMANDEUR(S) OU DU MANDATAIRE (Nom et qualité du signataire)		Le 15 juillet 2002 Maxime PETIT N°00.0407 RINUUY, SANTARELLI	

